

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

FINAL REPORT

7.9-10065

DESIGN OF A LOW COST EARTH RESOURCES SYSTEM

CR-150838

by

N. L. Faust
M. D. Furman

Engineering Experiment Station

and

G. W. Spann
Metrics, Inc.

"Made available under NASA sponsorship
in the interest of early and wide dis-
semination of Earth Resources Survey
Program information and without liability
for any use made thereof."

Contract No. NAS8-32397
GT/Project No. A-1938

Prepared for

National Aeronautics & Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

Prepared by

Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332

August 31, 1978



N79-16323

Unclass
00065

G3/43

CSCI 12E

(E79-10065) DESIGN OF A LOW COST EARTH
RESOURCES SYSTEM Final Report (Georgia
Inst. of Tech.) 161 F HC A08/MF A01

FINAL REPORT

DESIGN OF A LOW COST EARTH RESOURCES SYSTEM

by

**N. L. Faust
M. D. Furman**

Engineering Experiment Station

and

**G. W. Spann
Metrics, Inc.**

**Contract No. NAS8-32397
GT/Project No. A-1938**

Prepared for

**National Aeronautics & Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812**

Prepared by

**Georgia Institute of Technology
Engineering Experiment Station
Atlanta, Georgia 30332**

August 31, 1978

TABLE OF CONTENTS

I. INTRODUCTION.....	1
II. CAPABILITIES FOR AND USE OF DIGITAL LANDSAT DATA IN THE SOUTHEAST	4
Alabama.....	4
Georgia.....	4
Kentucky.....	9
Missouri.....	11
North Carolina.....	11
South Carolina.....	12
Tennessee.....	12
Summary.....	13
III. DESIGN OF LOW COST EARTH RESOURCES DATA PROCESSING SYSTEM.....	14
IV. THE IMGRID GEOGRAPHIC ANALYSIS PROGRAM.....	22
V. APPENDIX A DESCRIPTION OF MAJOR SOFTWARE RELATED TO IMAGE PROCESSING AND EARTH RESOURCES AT EES.....	30
RECRAW.....	31
RECCL.....	31
3DPLOT.....	31
MAX18.....	32
2DFFT.....	32
FILTER.....	33
IMGRID.....	33
TOPO.....	33
SCORECARD.....	34
THERMAL.....	34

PRECEDING PAGE BLANK NOT FINE

TABLE OF CONTENTS
CONTINUED

CHAN24.....	34
M6OCL.....	35
ASTEP.....	35
SEARCH.....	36
COORD.....	36
CLUSTER.....	37
CLUST.....	37
SUPERG.....	38
TRAIN.....	38
VI. APPENDIX B	
PROGRAM LISTING.....	39
ADDSIGM.....	40
ALARM (SUBROUTINE).....	42
ANGDIS (SUBROUTINE).....	43
ANGLE (SUBROUTINE).....	44
AUTO811Z.....	45
CHAN24.....	48
CLASIFY.....	51
CLEANUP.....	56
CLUSTA (SUBROUTINE).....	57
COMBINE.....	61
COPY5.....	63
COUNTY.....	64
CRDEM4.....	65
CYTAPE.....	66

TABLE OF CONTENTS
CONTINUED

DISDEM4.....	67
DRAWER.....	68
DTAPE.....	69
EBCDIC (SUBROUTINE).....	70
EDIST (SUBROUTINE).....	71
EIGEN (SUBROUTINE).....	72
ELIP.....	74
FACANL (SUBROUTINE).....	75
FILTER 1.....	76
GETPOLY.....	77
GSIG (SUBROUTINE).....	79
HISTO (SUBROUTINE).....	80
HISTPR.....	81
INITCL.....	83
INPUTSIG.....	84
LINBOX (SUBROUTINE).....	85
LINCLASS.....	86
LINEAR (SUBROUTINE).....	88
LISTALL.....	89
LISTSIG.....	90
MATIN (SUBROUTINE).....	91
MATPRT.....	93
MESS.....	94
MODIFY (SUBROUTINE).....	95

TABLE OF CONTENTS
CONTINUED

PACK (SUBROUTINE).....	96
POLY2 (SUBROUTINE).....	97
POLYCL	101
PUTIN.....	105
RAINBOW.....	106
RATIO.....	107
RDATA.....	108
RDING (SUBROUTINE).....	109
RETING (SUBROUTINE).....	110
SAVING (SUBROUTINE).....	111
SCORECARD.....	112
SEQCOV (SUBROUTINE).....	114
SEQST (SUBROUTINE).....	115
STRIPLESS.....	116
STRIPNF.....	117
SUMCAT.....	121
SYMINV (SUBROUTINE).....	122
THRDST (SUBROUTINE).....	123
TOPO2.....	124
TRAIN3.....	125
UNPAC1 (SUBROUTINE).....	130
UNPAC4 (SUBROUTINE).....	131
UPPLT (SUBROUTINE).....	132
ZCOUNTY.....	133

TABLE OF CONTENTS
CONTINUED

VII. APPENDIX C	
MAIN PROGRAM FLOWCHARTS.....	134
ADDSIGM.....	135
CLUSTER.....	136
LINCLASS.....	138
M60CL.....	140
STRIPLESS.....	142
STRIPNF.....	143
TRAIN3.....	145

I. INTRODUCTION

The Georgia Institute of Technology Engineering Experiment Station (EES) has been funded since 1973 by NASA Marshall Space Flight Center (MSFC) to assist the State of Georgia in utilizing Landsat digital analysis. In support of the Georgia state and local agencies and the usefulness of Landsat applications, the purpose of this project was to facilitate a transfer of technology, especially as related to low-cost Landsat data analysis systems. The major tasks accomplished during this project are discussed below.

The initial task involved a comprehensive survey of capabilities for digital processing of Landsat data in the Southern United States. Included was a review of the software and hardware currently being used by the southeastern states. Also, information was obtained on the commercial services used by those desiring to employ a Landsat data processing service.

As a second task, EES developed a set of minimum requirements (hardware and software) for a stand-alone Earth Resources Data Analysis System (ERDAS). The requirements developed in this phase represent a system that can be constructed at a minimum cost to the prospective user and can also satisfy the user's basic needs. The hardware for this system includes a minicomputer, line printer, disk drive, and tape drive while the software includes basic Landsat analysis techniques -- level slicing, maximum likelihood pattern recognition, clustering, and software scaling of the data. The primary products from such a system are computer printouts and statistics.

Since the user groups supporting an ERDAS system may vary greatly in size and available resources, a comprehensive plan was developed for expansion of the basic ERDAS system. This plan details a set of options, including approximate costs, and discusses the advantages of special equipment as well as the way that each equipment type complements the ERDAS concept.

One of the prime concerns in setting up an operational system is the hardware and software support needed to sustain a satisfactory operating schedule. Therefore, as the third task for this project, EES, through experience with its own ERDAS system, developed minimum re-

quirements for continuing support for such a facility (software and hardware). A system such as ERDAS is versatile and is likely to be used for purposes other than earth resources analysis. It therefore will be in use most of the time and will necessarily require the availability of a skilled electrical technician. Software support may be obtained "in-house" or by contract. Some equipment manufacturers have extensive service contracts which prove cost effective.

As the fourth task of the project, EES has assembled a software library for the digital processing of Landsat and other multispectral scanner data using, whenever possible, existing techniques from NASA, the University System, and other public agencies. Software techniques for maximum likelihood classification, linear classification, clustering, level slicing, registration and rectification, and table look-up classification are available for implementation into an ERDAS system. Each module is designed on a user keyword structure so that no detailed knowledge of programming is needed. Documentation is available on programs written or changed exclusively at EES with references provided for other programs. Even though the basic software was developed on a Data General minicomputer at Georgia Tech, the software library was designed entirely in Fortran IV and the routines are transferable to other 16 bit minicomputers.

The final task of the project was the implementation of a limited version of the IMGRID Geographic Analysis program on the ERDAS system. By providing this system to state and local users, the wide range of applications for rectified Landsat data becomes far more evident than if the system were geared to produce only land cover maps. IMGRID produces a dynamic modeling tool for site planning, erosion control, environmental impact, and many other uses.

Report Organization

The remainder of this report discusses each of the five tasks in detail. Section II discusses the results of the comprehensive survey of capabilities for digital processing of Landsat data in the Southeastern United States. Section III presents various options for low

cost earth resources processing systems and the minimum requirements for support of such facilities. Section IV discusses the IMGRID geographic analysis program. Appendix A contains descriptions of some of the major software available at EES, Appendix B contains a listing of the available Fortran software, and Appendix C presents flowcharts of some of the main analysis programs.

II. CAPABILITIES FOR AND USE OF DIGITAL LANDSAT DATA IN THE SOUTHEAST

Numerous individuals and/or organizations who are involved in the use of remote sensing data in the southeastern states were contacted in order to obtain information on their states' capabilities for digital processing of Landsat data. Included is a review of the software and hardware which are currently being used by these states as well as information on the commercial services used by those who wish to employ a Landsat data processing service.

The states involved are: Alabama, Georgia, Kentucky, Missouri, North Carolina, South Carolina and Tennessee. A brief summary of each state's capabilities also includes information on state funded projects utilizing Landsat data digitally processed by companies or organizations outside the state.

Alabama

NASA/Marshall Space Flight Center (MSFC) has the only current capabilities in Alabama for digital processing of Landsat data. Their hardware system consists of an IBM 360/75 with two megabytes of memory and several discs and tapes and a PDP11/45 with two 250-mega-byte discs, three tape drives, and various displays and terminals. Approximately 375 computer software routines are available including all algorithms developed by any of the NASA centers. However, very little analysis of Landsat tapes is actually done with this system at present.

Georgia

In order to assist the State of Georgia agencies with their desire to incorporate digital Landsat data into their planning activities on an operational basis, the Georgia Tech Engineering Experiment Station (EES) approved the design and acquisition of the Earth Resources Data

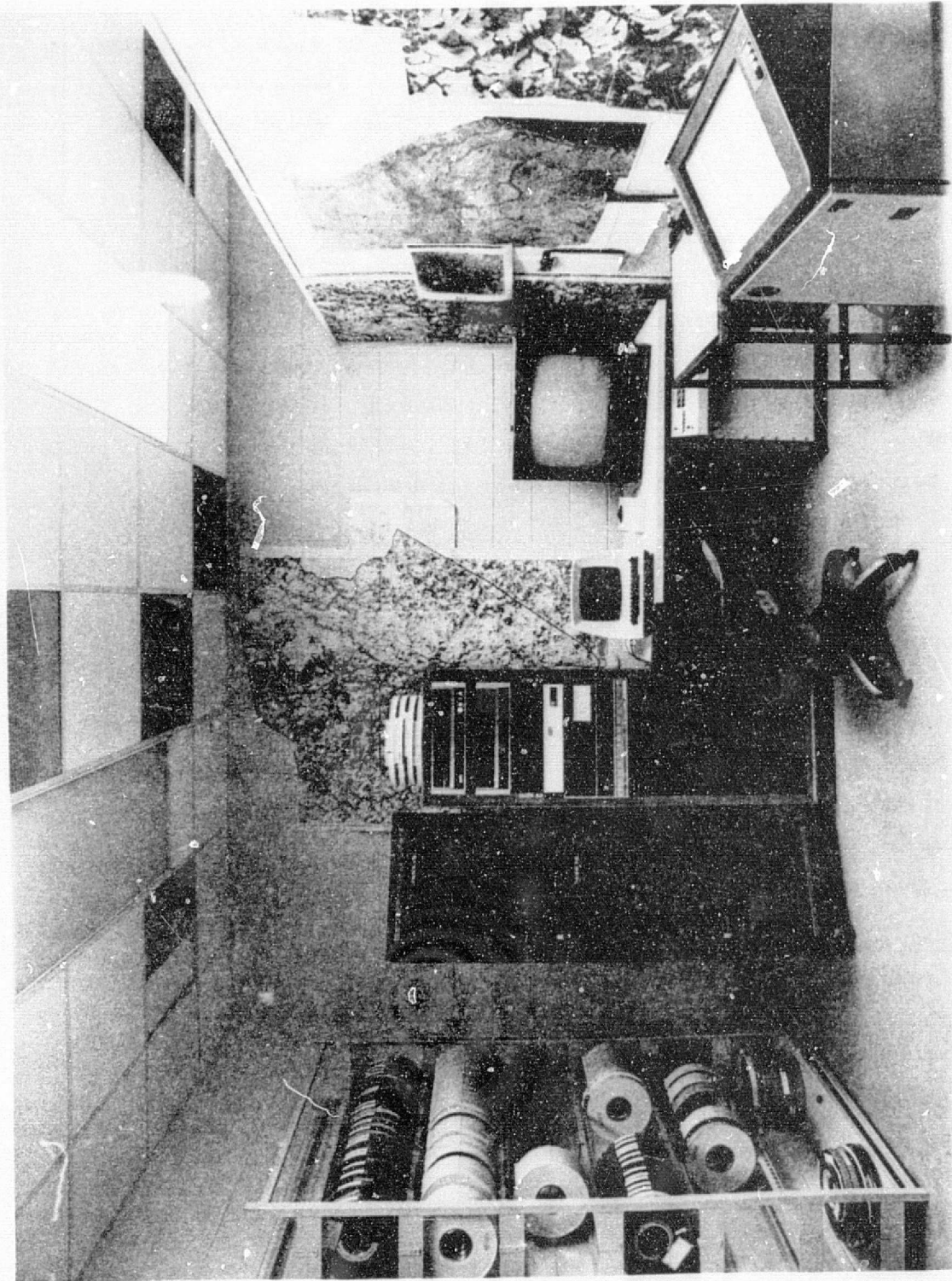


Figure 1. Georgia Tech ERDAS System

ORIGINAL PAGE IS
OF POOR QUALITY

Analysis System (ERDAS). EES was responsible for selecting and integrating both the hardware and software components of the system.

Hardware. ERDAS was designed and constructed to allow true interactive digital processing of all types of remote sensing data. The system consists of a set of four modules: (1) minicomputer subsystem, (2) input medium, (3) hardcopy output medium, and (4) display subsystem.

The minicomputer subsystem consists of a NOVA-2/10 minicomputer with 64K bytes of core memory and a dual Diablo disk system with 5.0 megabytes of storage for programs or data.

The input medium for the ERDAS system is a set of two nine track dual density (phase encoded/NRZI selectable) magnetic tape drives and controller -- both drives with a capacity for 10-1/2 inch reel tape.

One hardcopy output device is a twenty inch electrostatic dot matrix printer/plotter. Scaled maps of Earth Resources data can be made using this device. A CROMALIN^(R) photographic process may then be used to generate a color coded output hardcopy product. Another output method currently in use consists of storing images on a magnetic tape and sending these tapes to be made into images by the use of a digital film writer. This method is currently very inexpensive.

The display subsystem consists of a high quality color video image analysis system that is interfaced to the minicomputer for complete user interaction in the choice of training samples for earth resources classification.

Software. Initially, EES implemented a basic Landsat digital analysis program called ASTEP (Algorithm Simulation Test and Evaluation Program) which was obtained from NASA/JSC and has been extensively modified by EES personnel. The ASTEP system was designed as a modular program whereby various classification algorithms may be tested against one another using a standard input/output system.

Software for the rectification of Landsat data to map coordinates using a least squares fit of Landsat data to control points was obtained from NASA/MSFC and transferred to the Georgia Tech computer. Other software, such as various spatial clustering algorithms, was studied but has not yet been transferred. Software for a table lookup formulation for Landsat classification (ELLTAB) was obtained from NASA/ERL along with software for rectification, destriping, and polygon location of Landsat data. A fast combination table lookup and maximum likelihood classifier from NASA/ERL has been implemented which significantly decreases the length of time needed for a scene classification. A fast clustering algorithm from NASA/ERL has also been implemented. In addition, many software algorithms for image manipulation, spatial filtering, rectification, training field selection, and high speed classification have been developed at EES. EES software exists for image analysis on UNIVAC and CDC large computers and Data General minicomputers.

Projects. EES has been deeply involved in the formulation, planning, and implementation of a Georgia Natural Resources Inventory since its conception in 1972. During 1972 and 1973, EES presented various State of Georgia agencies with the background information needed to make an initial assessment of the usefulness of digital Landsat information. A trial project was initiated between the Georgia Department of Natural Resources (DNR) and EES in 1973 to test the capability for using digitally processed Landsat data to determine land use in the Atlanta area.

EES has been funded since 1973 by NASA/MSFC to assist the State of Georgia in utilizing Landsat digital analysis for various resource problems within the state. In this multi-year effort, several related tasks have been performed in conjunction with numerous local and state agencies within Georgia, including the Department of Natural Resources, the Department of Transportation (DOT), and the Office of Planning and Budget (OPB).

In 1975 Georgia Tech EES was designated as the technical interface with NASA/ERL for the transfer of NASA software to the State of Georgia. Under this technology transfer project EES purchased the necessary digital processing equipment required for operational processing of Landsat data. EES then coordinated a project with the Georgia DNR office of Planning and Research for classifying and mapping land cover for the State of Georgia. Products of the effort assured further acceptance of digital processing of Landsat data as an operational tool for environmental analysis.

Probably the best indicator of the success of the technology transfer efforts of this project is the number of agencies which have committed funds and/or personnel time to a project to map the entire State of Georgia using Landsat data. This project is concerned with mapping land cover using Landsat data processed with ERDAS and, where appropriate, inferring land use.

The agencies which have committed funds to the mapping project include:

- Georgia Department of Natural Resources
 - Environmental Protection Division
 - Game and Fish Division
 - Office of Planning and Research

- Georgia Forestry Commission

- Georgia Office of Planning and Budget
 - Bureau of Community Affairs

- U.S. Department of Agriculture
 - Soil Conservation Service
 - Forest Service

- United States Army Corps of Engineers
 - Fort Benning
 - Savannah Engineer District

- North Georgia Area Planning and Development Commission

- Coosa Valley Area Planning and Development Commission

Other organizations which are interested but as yet have supplied no funds include:

- Georgia Department of Natural Resources
 - Earth and Water Division

- Georgia Department of Transportation

- Five other area planning and development commissions.

EES is currently assisting the Georgia DNR in a geographic data base demonstration project. DNR manually obtained 30 different data variables on a 10 acre cell basis for North Fulton County, Georgia. EES is assisting with transfer of the data to the ERDAS system and analysis of the data using IMGRID and NIMGRID.

In addition to the activities at Georgia Tech EES, the Atlanta Regional Commission (ARC) was involved in a project to test the feasibility of using automatic processing of Landsat data to detect land use changes in the seven county planning area of the ARC, and thereby update the USGS/LUDA land use map of the Atlanta Region.

The technique chosen to accomplish the land use change detection was the ratioing of two different dates of Landsat data. This was accomplished at the EROS Data Center utilizing the Image 100 System. The results of the project indicated that ratioing Landsat data was a feasible technique for ARC to use in updating the USGS land use maps of the area. Accuracy evaluations showed that 91% of the change theme was accurate to within about 3 pixels (accuracy sufficient for ARC's purposes). Additional manual analysis was required to identify accurately the types of land use changes.

Kentucky

Kentucky is currently developing the capability for in-house processing of Landsat digital tapes. A few state agencies and universities have small interactive graphics systems which are capable of digital processing, but these are being discarded in favor of a central state computer system. The central system includes an IBM 370/168 MP with 9 megabytes of main storage and various discs. The Bureau of Computer Services has been established as the central hardware agency. Their personnel have recently visited NASA/ERL to obtain compatible software for the system.

A representative from Eastern Kentucky University recently attended a workshop at GSFC on a joint Appalachian Regional Commission-NASA project to apply Landsat to the study of geological lineaments using GE's Image 100 System. An ORSA package was ordered from GSFC which provides an offline printout of similar pixels vs. significant changes, but this is not adequate for final classifications.

Five professors from Murray State University attended a short course at ERL and had plans to return the end of October for hands-on experience with digital tapes for strip mining studies in Western Kentucky.

Previous projects in Kentucky include a survey in 1973 by Earthsat of water impoundments larger than two acres. The results were not beneficial because the imagery was taken after a flood and strip mines appeared as impoundments. However, this led to a Landsat-2 follow-on study in 1976 for which ERIM provided the digital processing. The objective of the project was to look at the operations of inspection and enforcement to detect significant violations of surface mining laws.

Several projects have been contracted with Bendix Corporation for Area Development (AD) Districts in conjunction with EPA 208 land use planning programs. Color-coded maps and overlays were provided in 1976 for the Kentuckiana AD (seven Kentucky counties, two Indiana counties) at 1:48,000 scale. Area tabulations were also provided for each county. Processing for the Big Sandy AD (Prestonsburg, Kentucky) is in progress and maps are scheduled for delivery this fall. A contract has also been signed for the Green River AD (Owensboro, Kentucky) to be delivered in December, 1977. This project will provide information to meet HUD 701 requirements for land use planning.

Two counties of Kentucky were included in a digital processing project by Bendix for the Ohio-Kentucky-Indiana Council of Government (OKI COG). Color-coded maps at 1:62,500 scale and computer tabulations for each of 229 drainage basins were provided. Also, five counties centered around Fayette County (Lexington, Kentucky) known as the

Central Blue Grass Region (not an AD district) were mapped for the Army Corps of Engineers.

The LARS program at Purdue was used to produce a land cover map of Henderson County in Western Kentucky (part of Southeastern Indiana COG).

Missouri

According to the final report of a project on Earth Observation Data Management Systems in December, 1976, "few agencies (in the five state Midwestern region of Illinois, Iowa, Minnesota, Missouri, and Wisconsin) now have the staff or computer capabilities to handle digital satellite data."¹ At the present time, Missouri still has no capabilities for digital processing of Landsat data. However, the University of Missouri at Rolla recently received a grant from the National Science Foundation and has ordered a Comtal Interactive Image Analysis System which will be used for such processing. Delivery occurred in early 1978. Software will be obtained from NASA/ERL.

Several Missouri agencies have funded projects for digital processing in the past. A demonstration study of land use in the Ozarks Planning Region of Southern Missouri was conducted by NASA/ERL in 1975.² The Soil Conservation Service in Missouri has digital tapes processed by LARS/Purdue for water analysis and study of soil patterns. Also, the Missouri Geological Survey had the University of Kansas in Lawrence process some digital tapes for an area around Kansas City, Missouri.

North Carolina

Currently North Carolina has no facilities for processing digital Landsat data although considerable interest exists for establishing a centralized state system. Several projects have been completed by Bendix Corporation for EPA 208 planning regions in North Carolina. Color-coded land use overlays at 1:96,000 scale were produced using the Multispectral Data Analysis System (M-DAS) at Bendix in 1975 for

¹Eastwood, et al., "Project on Earth Observation Data Management Systems," Final Report, Washington University, St. Louis, Missouri. Prepared for CSFC, December 31, 1976.

²"A Computer Implemented Land Use Classification Technique Applied With ERTS Digital Data Acquired Over Southern Missouri," Report number 143, April 1975, ERL/JSV, A. T. Joyce and J. D. Derbonne'.

Planning Region J (also known as Triangle J -- five counties in central North Carolina -- Raleigh, Durham, Chapel Hill areas). An additional product of the analysis process was the generation of statistical data by 50x50 meter grid cells in data sets corresponding to 54 7-1/2 minute USGS quadrangle maps. Overall classification accuracy of the land use categories was judged to be around 90% and the cost of processing the Landsat scene and generating the products was approximately \$4.00 per square mile.

A similar analysis was done in 1976 by Bendix for Planning Region D in the Northwestern part of the state. Also, analysis of the Dan River Sub-Basin (Roanoke River Basin) was completed this year for the Corps of Engineers. NASA had some involvement in the most recent project.

South Carolina

South Carolina has no capabilities at present for digital processing of Landsat data. A proposal has been submitted to NASA/ERL for projects which may involve digital processing. Previous activities include a contract with General Electric Company for Image 100 processing of Landsat data to produce color-coded land use maps and area calculations for three Council of Government (COG) regions comprising approximately 25% of the state. Landsat tapes were processed by NASA to produce a map of the Congaree Swamp area for the Wildlife and Marine Resources Department. The Land Reserve Conservation Commission and the Bureau of Mines visited the EROS Data Center for processing of digital tapes for a study of mining areas in South Carolina.

Tennessee

The only current capabilities in Tennessee for digital processing of Landsat tapes are at the University of Tennessee, Knoxville. The facilities are presently being used for image processing projects other than earth resources, although previously they have been used for Landsat analysis. Oak Ridge National Laboratory has accomplished strip mining surveys using some Landsat data.

Bendix Corporation completed a digital processing project on water resources and strip mining in the New River Drainage Basin in North-central Tennessee for the Soil Conservation Service in early 1976. They provided a color-coded map of the Basin at 1:62,500 scale, fifteen color-coded 7-1/2 minute quads, computer tabulations, and rescanned-resampled tapes for the area.

Summary

Of all the Southeastern states, Georgia is the most advanced in the use of digital Landsat data. The University of Tennessee at Knoxville has the capability for digital processing of Landsat data but no such projects are currently underway. All other states have relied on NASA or commercial facilities. The most extensive use of digital Landsat data among these states has been in fulfillment of EPA 208 and HUD 701 planning requirements.

The results of this survey indicate that users of remote sensing data in the Southeastern U.S. are increasingly turning to digital processing techniques. All the states surveyed have had some involvement in projects using digitally processed data. Even those states which do not yet have in-house capabilities for digital processing are extremely interested in and are planning to develop such capabilities.

III. DESIGN OF LOW COST EARTH RESOURCES DATA PROCESSING SYSTEMS

In the design of an earth resources data processing system there are many factors to be considered. In some cases potential users should buy turn-key systems that are currently on the market; in other cases, users might consider the design and implementation of their own systems by buying components and assembling the systems using their own technical expertise. This section deals with alternative systems that might be considered by users having the technical expertise for assembly of such a system within their agencies or support groups.

With the appropriate technical personnel available, a significant cost savings often may be realized by user design and implementation of systems. At least one part- or full-time computer hardware technician, one applications software analyst, and several applications programmers are desirable for all phases of system design. These requirements may be lessened, however, if sufficient support is available from the various equipment manufacturers.

Computer software for digital processing of earth resources and other geographically based data is currently becoming available for minicomputers at minimal or no cost. Since a significant amount of this computer software has been developed under government contracts, it is in the public domain and readily available. Thus the costs involved in the acquisition of the systems listed here are primarily the actual hardware costs. If they are needed, personnel training costs and systems interface costs are extra.

Figure 2 indicates five alternative configurations for low cost earth resources data processing systems. At the low end is a nominal system consisting of a minicomputer, floppy disk, magnetic tape unit, color terminal, and line printer. The estimated cost range for this system (depending on the exact components selected) is \$22,000 to \$45,000.

At the upper end of the range (still, however, at a price significantly less than many systems on the market) are systems costing an estimated \$165,000 to \$220,000. This configuration, with substantially increased capability over the lower cost system, consists of a mini-computer with an array processor, a 96 megabyte disk, dual magnetic tape drives, a digitizer, a color display, a line printer, and a film recorder.

In general, as the systems increase in cost, the processing sophistication is improved and the speed with which a data set can be analyzed increases rapidly. Thus, for users requiring only a low volume of processed data, a system in the low or middle cost range might be suitable. For users desiring a faster processing speed and an increased throughput, a more expensive system might be in order.

The breakdown of the total costs for each system is shown in Table I. For each system, the high and the low cost estimate for each component is given. Thus, by selecting particular components with a greater or lesser capability, systems could be configured that cost anywhere within the range of the cost extremes given.

To facilitate the estimation of costs for system configurations other than those listed in Figure 2, the costs of the individual components are given in Table II. Using these data, the approximate costs of many more low cost system configurations could be derived. Typical vendors of such components are given in Table III.

Estimated Cost for ERDAS Support

A minimum cost estimate for the support of the Georgia Tech ERDAS system over a one year period totaled \$2,500. This estimate includes approximately 400 hours of a resident part-time technician along with all electrostatic line printer supplies and any additional maintenance charges incurred when repairs exceeded in-house capabilities.

This estimate is approximately 3.3% of the total cost of the ERDAS system. As maintenance contracts usually run about 10% of system costs per year, a significant savings was realized using in-house methods of maintaining the equipment.

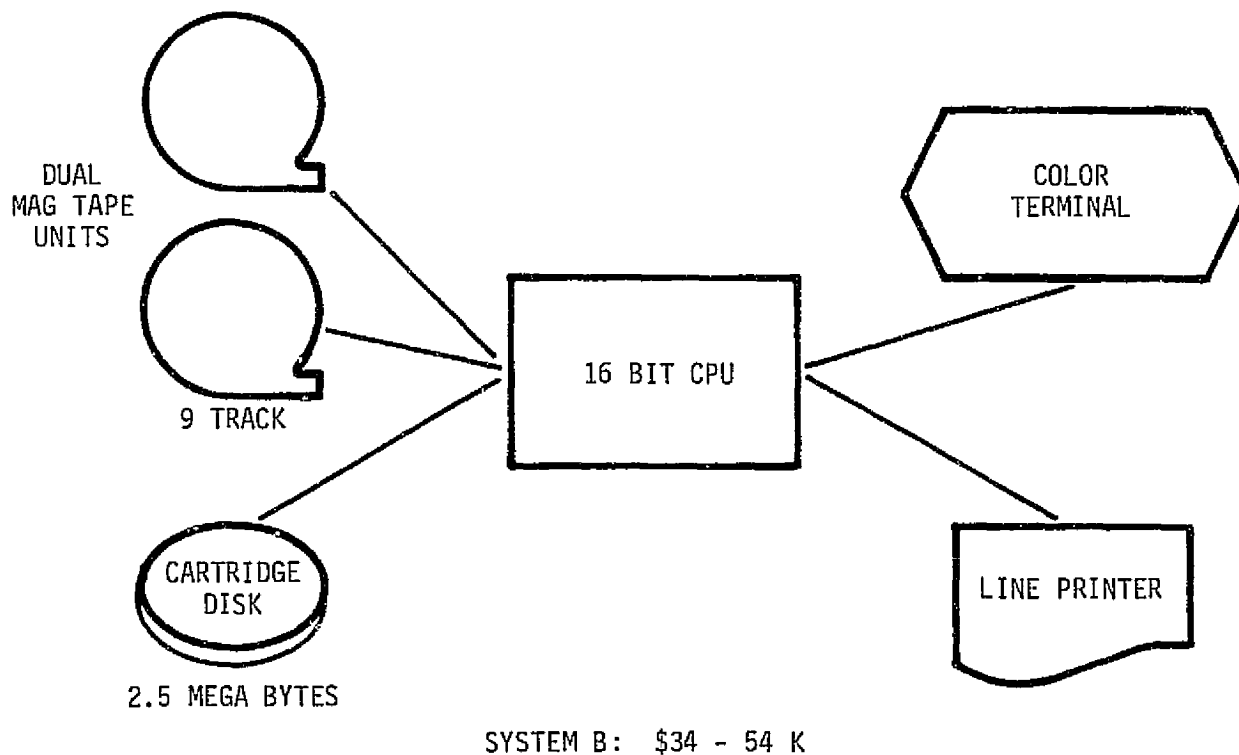
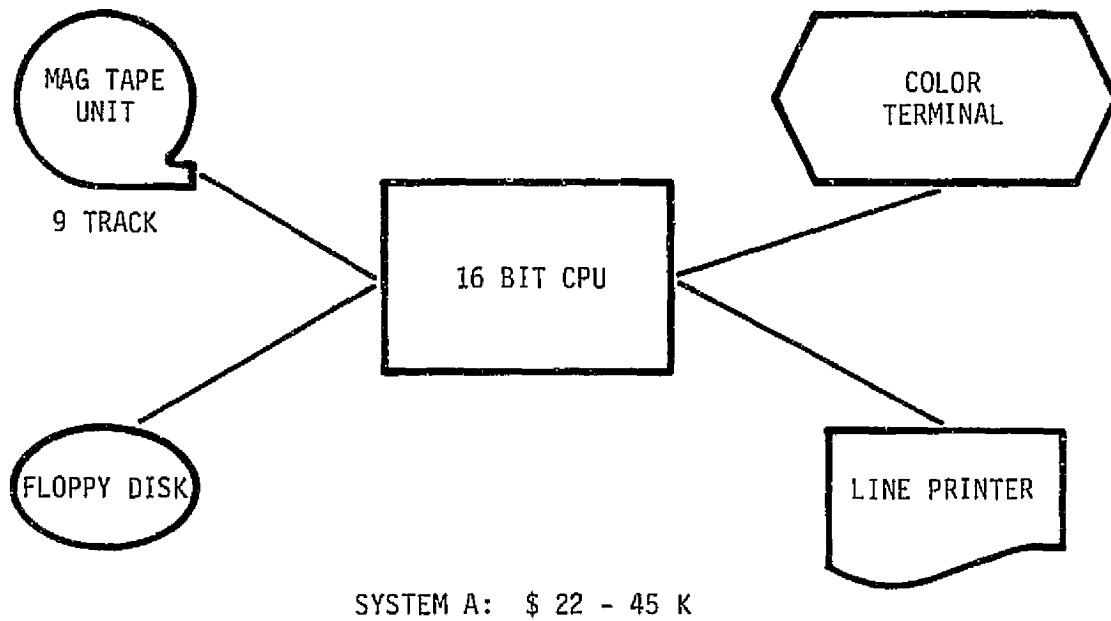


Figure 2

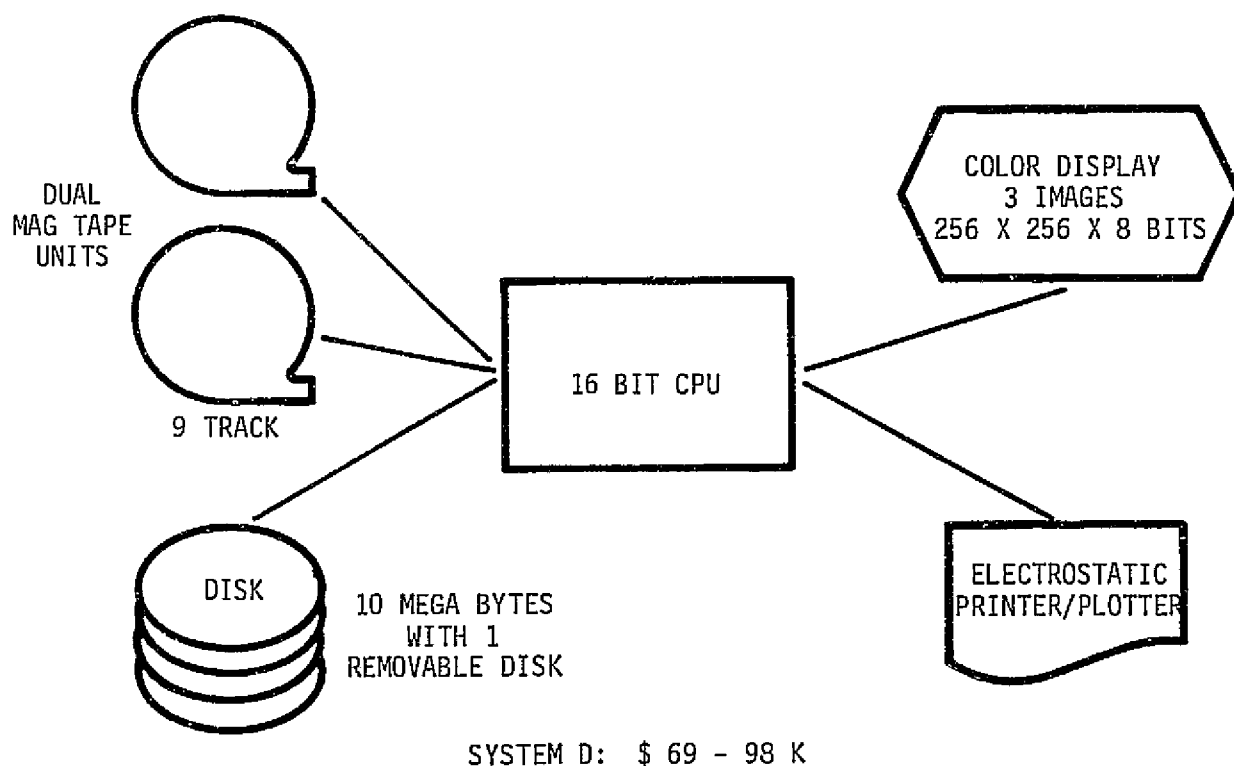
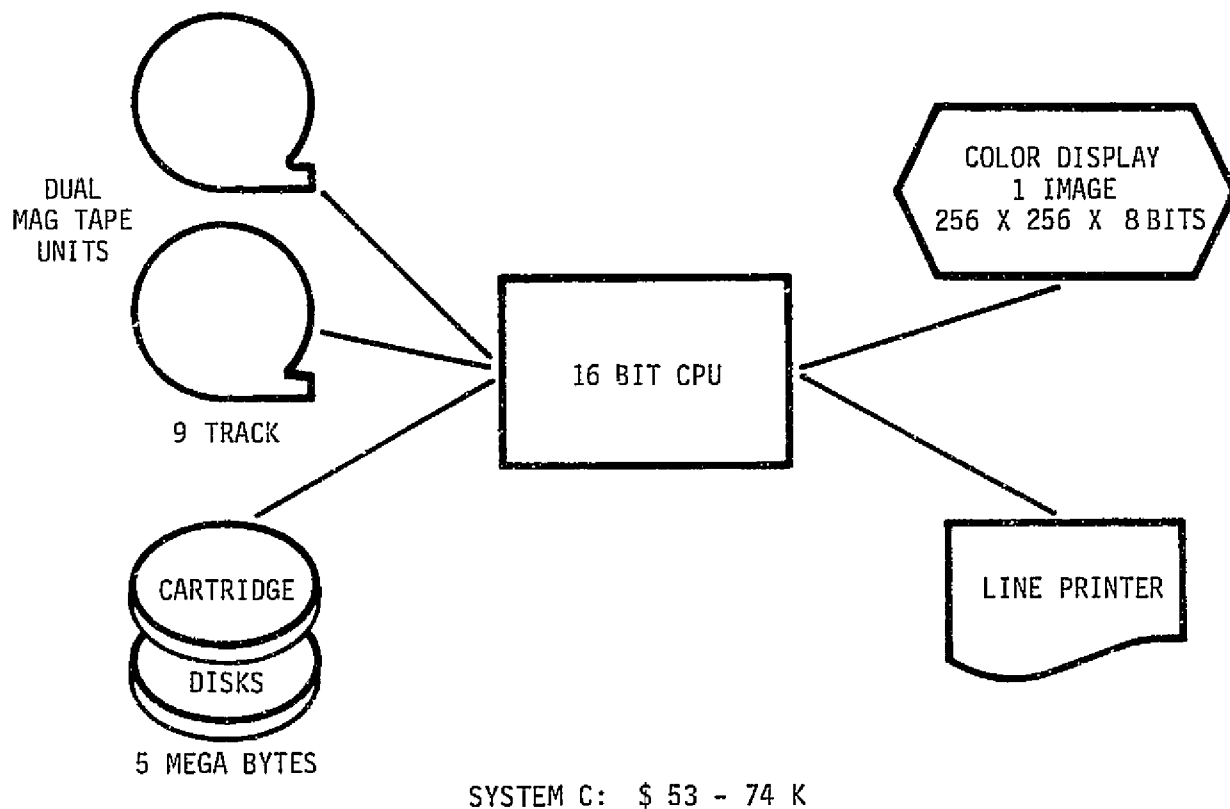
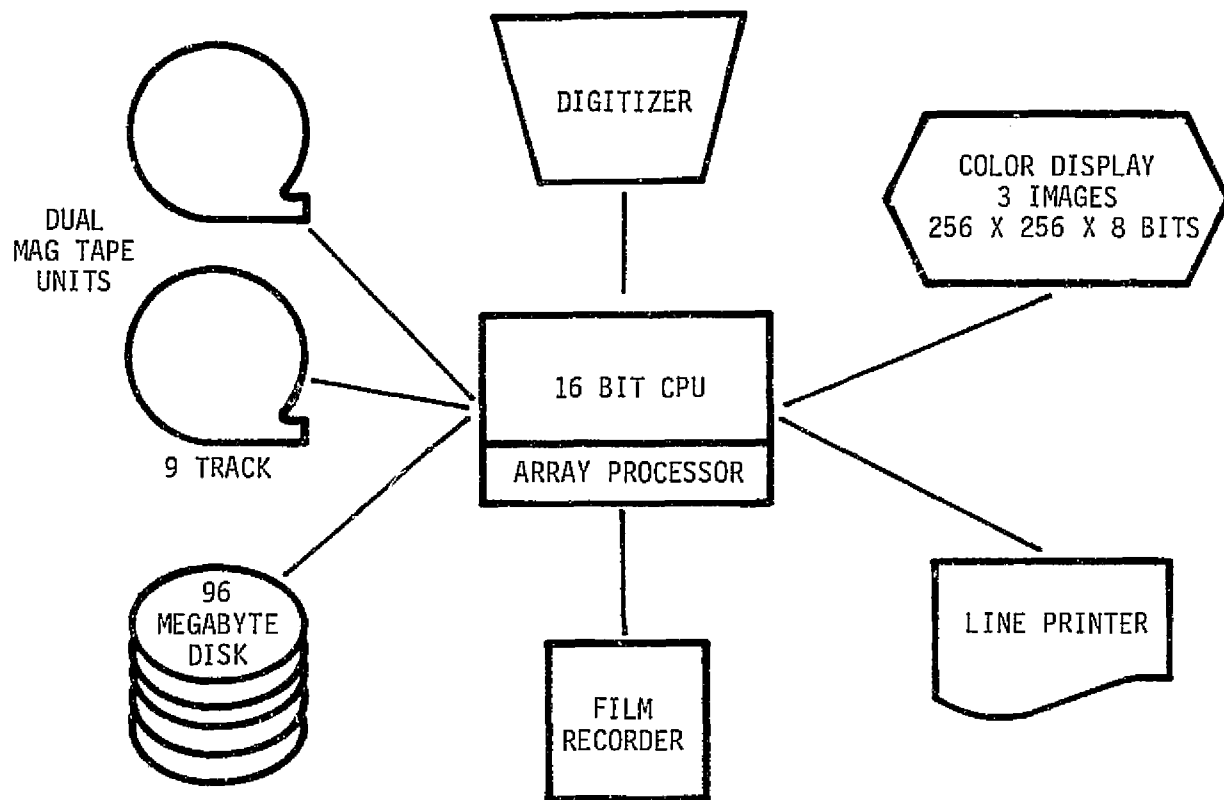


Figure 2 (Continued)



SYSTEM E: \$165 - 220 K

Figure 2 (Continued)

TABLE I. INDIVIDUAL SYSTEM COMPONENT
COST ESTIMATE

		LOW - HIGH
SYSTEM A:	TAPE DRIVE	\$ 5 - 10 K
	CPU	\$ 8 - 18 K
	DISK	\$ 2 - 4 K
	LINE PRINTER	\$ 5 - 8 K
	COLOR TERMINAL	\$ 2 - 5 K
		<u>\$ 22 - 45 K</u>
SYSTEM B:	TAPE DRIVE	\$ 12 - 14 K
	CPU	\$ 8 - 18 K
	DISK	\$ 7 - 9 K
	LINE PRINTER	\$ 5 - 8 K
	COLOR TERMINAL	\$ 2 - 5 K
		<u>\$ 34 - 54 K</u>
SYSTEM C:	TAPE DRIVE	\$ 12 - 14 K
	CPU	\$ 8 - 18 K
	DISK	\$ 9 - 11 K
	LINE PRINTER	\$ 5 - 8 K
	COLOR DISPLAY (256 X 256)	\$ 19 - 23 K
		<u>\$ 53 - 74 K</u>
SYSTEM D:	TAPE DRIVE	\$ 12 - 14 K
	CPU	\$ 8 - 18 K
	DISK	\$ 9 - 11 K
	ELECTROSTATIC PRINTER/PLOTTER	\$ 10 - 20 K
	COLOR DISPLAY (3 images 256 X 256)	\$ 30 - 35 K
		<u>\$ 69 - 98 K</u>
SYSTEM E:	TAPE DRIVE	\$ 12 - 14 K
	CPU	\$ 8 - 18 K
	DISK	\$ 20 - 35 K
	FILM WRITER	\$ 40 - 50 K
	ARRAY PROCESSOR	\$ 40 - 45 K
	LINE PRINTER	\$ 5 - 8 K
	DIGITIZER	\$ 5 - 15 K
	COLOR DISPLAY (3 images 256 X 256)	\$ 30 - 35 K
		<u>\$165 - 220 K</u>

TABLE II. INDIVIDUAL COMPONENT COST ESTIMATE

1)	1 Mag Tape Drive & Controller	\$ 5 - 10 K
2)	2 Mag Tape Drives & Controller	12 - 14 K
3)	16 Bit CPU with 32 K Memory	8 - 18 K
4)	Line Printer	5 - 8 K
5)	Floppy Disk	2 - 4 K
6)	Dual 2.5 Megabyte Drive & Controller	9 - 11 K
7)	10.0 Megabyte Cartridge Disk	9 - 11 K
8)	Electrostatic Printer/Plotter	10 - 20 K
9)	Array Processor	40 - 45 K
10)	96 Megabyte Disk	25 - 35 K
11)	Color Terminal	2 - 5 K
12)	Color Display Image 256 x 256	19 - 23 K
13)	Color Display 3 Image 256 x 256	28 - 35 K
14)	Film Writer	40 - 50 K

TABLE III. TYPICAL VENDORS

COMPUTERS

Data General Corporation
Digital Equipment Corporation
Hewlett Packard Corporation

MAGNETIC TAPE DRIVES

Digi-Data Corporation
Pertec Computer Corporation
Kennedy, C. J. Company

DISKS

Control Data Corporation
Data General Corporation
Digital Equipment Corporation

COLOR DISPLAY SYSTEMS

Aydin Corporation
Comtal Systems Corporation
ITT Grinnell Corporation
Ramtek

ELECTROSTATIC PRINTERS/PLOTTERS

Versatec Incorporated
Varian Data Machines
Gould Incorporated

DIGITAL FILM RECORDERS

Dicomed Corporation
Optronics International Incorporated

DIGITIZERS

Summagraphics
Bendix Corporation
Talos System Incorporated
Aristo Graphics Corporation

LINE PRINTERS

General Electric Company
Okidata Corporation
IBM
Varian Data Machines

IV. THE IMGRID GEOGRAPHIC ANALYSIS PROGRAM

IMGRID, an information manipulation system for grid cell data structures, is a package of computer programs designed for the analysis of natural resource and land planning data which is qualitative in its thematic content and varies over geographic space. Originally developed by David Sinton of the Harvard University Department of Landscape Architecture, the IMGRID system has been adapted for use on mini-computers at Georgia Tech.

The IMGRID system has been designed for people having no previous experience with computers. The basic operations are controlled with simple keyword commands which may be used with a basic knowledge of planning principles but without any knowledge of programming. Thus, a link is provided for easy access and manipulation of digital data bases. Some typical project applications of the IMGRID system are:

- River basin planning
- Siting of facilities such as airports
or sanitary landfills
- Environmental impact statement review
- Visual Analysis
- Project review by regional planning agencies

KEYWORD STRUCTURE

The basic structure for an IMGRID keyword command involves three processes:

1. Retrieve one or more data elements from the data file.
2. Transform or manipulate the values for each grid cell
in the data elements retrieved.
3. Store the new data element created in the data file.

On the Georgia Tech ERDAS, keyword commands are entered on a CRT terminal, executed by the NOVA II mini-computer, and output either in color on the Comtal video display, or in black & white on a dot matrix printer. The data files are usually stored on tape then read into a disk file. The IMGRID program itself also resides on a disk.

The keyword commands fall basically into 9 groups:

1.) Data Entry & Management

Two keywords can be used to enter new information into the data base:

- STORE - operates on an element (data variable) by basis. When a new variable, such as a soil type or slope category is added to the data base for all cells, STORE can be used.
- UPDATE - operates on a cell by cell basis. If a single cell changes characteristics for a particular variable, such as a land use change from agricultural to residential, UPDATE is used along with the row and column location, plus the new value of the cell.

Three keywords which are used for data management are:

- RELOC - permits a data element to be moved to a new location within the data file, such as relocating the results of an analysis as a new data element in another location.
- RENAME - allows the name of a data element to be changed without affecting the contents of the data.
- LIST - allows a user to list the names associated with the contents of part or all of the data base.

2.) Delimiter Keywords

- MODEL - used as a first keyword in a sequence of keywords defining an analysis. The primary function of MODEL is to assign a title to subsequent analyses. It also includes the function of the CLEAR keyword.
- CLEAR - clears the results of operations performed by previous keywords.
- END - identifies the completion of IMGRID input.

3.) Display Keywords

- SYMBOL - allows a user to enter a specific set of character symbols to be used when making a map display.
- MAP - makes a graphic display of the contents of a data element in the file. This can be a data variable such as a MAP of a slope, or of the results of an analysis such as vulnerability to soil erosion.
- TEXT - permits user to insert textural descriptions of the procedures being undertaken.

4.) Spatial Analyses

- SEARCH - generates a set of values which identify the proximity of each cell in the study area to a specified condition, such as roads, rivers, or airports. The analysis determines how far every other location is from the preselected data items.
- ASEARCH - operates on a cell by cell basis examining for a prespecified radius, the conditions around every cell, or a defined subset of cells in the study area, such as how many cells of wetlands are there within a five cell radius of a landfill.

5.) Rescaling or Restructuring of Data Values

- RECODE - assigns new values to an old set of values for a data element, such as recoding an old set of values for element "land market value" to reflect land use or tax policy changes. Recode assigns values in the range of 0-9.
- XRECODE - an extended recode which assigns values in the range of 0-19. The RECODE keywords assign the last value specified in the case where multiple elements are being combined with different value scales.

- OVERLAY - assigns the highest value, rather than the last value, when multiple elements are being combined. The system will over-ride the values assigned on the first element if the new values generated by the rescaling are greater than the previously existing values.

6.) Logical Combinations of Elements

- MATRIX - results in a series of values which identify each of the possible combinations of features, such as the highly erodible soils on steep slopes.

7.) Reject Conditions

- REJECT - permits the user to identify a group of cells which must be ignored ("dropped out") in all analysis and display of the data, such as rejecting unstable soils in an analysis for siting a large industrial complex.

8.) Mathematical Manipulation of the Data

- MULTPLY - allows user to multiply the values in one data element by the values in a second data element.
- INDEX - generates a weighted index of several data elements and also provides for addition and subtraction.
- NORMAL - takes data values over large ranges and normalizes them to the range of 0-99.
- REDUCE - generates values in the range of 0-19.

9.) User Written Fortran Subroutines

- USERSUB - allows user to write a Fortran subroutine to perform any set of computations that is desired and call that subroutine through the USERSUB keyword.

Figure 3 shows an example of IMGRID output where elements of slope, depth to water table, soils, vegetation, proximity to roads, and travel time were weighted, overlaid, recoded and mapped to produce an attractiveness model for industrial parks. The darkest cells are best for industrial sites using the criteria specified.

050
049
048
046
047
045
044
043
042
041
040
039
038
037
036
035
034
033
032
031
030
029
028
027
026
025
024
023
022
021
020
019
018
017
016
015
014
013
012
011
010
009
008
007
006
005
002
001

26

[illegible]

Because of the large size of IMGRID and the limited address space of a 16 bit computer, a method was devised to partition the IMGRID program into independent subroutines. Each subroutine corresponds to a keyword in the above discussion. By making each major subroutine an overlay, IMGRID was implemented on the NOVA 2 minicomputer. A core part of the IMGRID program resides in the computer at all times, and when a particular keyword is selected, only programs that provide that function are loaded into the main memory of the minicomputer from disk storage. While some delay occurs in the program execution due to the overlay procedure, the time required is negligible compared to the execution time of the analysis modules.

The structure of the minicomputer IMGRID program allows two data variables in a format of up to sixty elements by sixty elements to be analyzed at one time. In the Atlanta, Georgia area, this array size is sufficient to represent data for a USGS seven and one half minute quadrangle in approximately ten acre elements. For a large area, therefore, data for a data set of the whole area would have to be partitioned into segments dependent on the selected cell size.

All keyword functions with the exception of the search algorithms could easily be performed on such a segmented data set. Any application for which a search from some criterium is specified, however, would encounter problems when needing to search beyond the boundaries of one data segment. In addition, the normal mode of operation for IMGRID is execution in a batch runstream in which a set of input data is required in special formats. While this method is sometimes desirable when using cards as input to a large computer, an alternate interaction method should be considered when dealing with minicomputers where the user has direct contact with the computer itself and he may often be the only user using the system at one time.

In an effort to alleviate some of the difficulties given above a new minicomputer version of IMGRID, called NIMGRID, has been developed at EES. NIMGRID reflects to a large extent the program philosophy of the original IMGRID program regarding types of functions performed on

the data base, but a new philosophy of data storage and retrieval and program interaction was implemented in early 1978. The data storage philosophy envisions each data variable as being represented by a raster data set which is only constrained by the disk space available to the user of a particular system. The data are processed on a line by line basis, and even for the search algorithm, only two lines of the data set are needed in the computer at one time. While this method involves many more input/output operations than the IMGRID method, very little degradation in performance has been noted in comparing NIMGRID and IMGRID.

The prime advantage of this storage and retrieval philosophy is that the size of the data set that may be processed by a user is virtually unlimited. Also, searches may be made over large areas without problems associated with crossing data segment boundaries. Another virtue of the NIMGRID system is that the program has been made interactive. Prompting of the user occurs, giving him all possible choices and asking him to select the desired command. No knowledge of FORTRAN and very little knowledge about the particular minicomputer is required of the user. Although NIMGRID is still under development, applications personnel from the Georgia Department of Natural Resources are currently using the program at EES with ease.

APPENDIX A
DESCRIPTION OF MAJOR SOFTWARE
RELATED TO IMAGE PROCESSING AND
EARTH RESOURCES AT EES

PROGRAM NAME: RECRAW
LANGUAGE: FORTRAN IV
COMPUTER: SEL 32/55, CDC Cyber 74

SPECIAL PERIPHERALS: 2 Tape Drives
Disk Files

PURPOSE: RECRAW uses a first order transformation matrix from COORD to resample raw Landsat data by a selectable nearest neighbor or bilinear interpolation to format the data into a standard coordinate system.

PROGRAM NAME: RECCL
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2, CDC Cyber 74, SEL 32/55
SPECIAL PERIPHERALS: 2 Tape Drives
Disk Files

PURPOSE: RECCL uses a first order transformation matrix computed by COORD to resample Landsat classified data using nearest neighbor and format the data into a Latitude-Longitude or UTM coordinate system.

PROGRAM NAME: 3DPLOT
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2, CDC Cyber 74
SPECIAL PERIPHERALS: Calcomp plotter or printer/plotter
Disk Files

PURPOSE: 3DPLOT provides a perspective view of three dimensional data ($z = f(x,y)$). Viewing angle and scaling may be specified by the user. This technique is especially useful in analysis of topographic related data.

PROGRAM NAME: MAX18 (NASA/ERL)
LANGUAGE: FORTRAN IV
COMPUTER: SEL 32/55, CDC Cyber 74
SPECIAL PERIPHERALS: Two Tape Drives
Disk Files

PURPOSE: MAX18 was developed by Ronnie Pearson of NASA/ERL (Slidell, Louisiana) as a fast classifier of Landsat MSS data. The program is a combined table look-up-maximum likelihood type of classifier which uses the best points of each technique. Instead of creating a look-up table defining the boundaries of statistical distribution of signatures, this technique iterates quickly through the data building a table of where in channel space the majority of the data in a Landsat scene lie. Then, these vectors are classified using a maximum likelihood scheme. A second iteration is made through the data for classification. Each pixel of MSS data is checked to see if the data vector associated with that pixel is in the already classified data table. If so, the classification is derived by simply indexing into the classified table. If the data vector has not already been classified a maximum likelihood decision rule is used. There is a distinct trade off between amount of storage and speed of classification. On the SEL machine a Landsat scene may be classified into 60 classes in 1.5 - 2 hours.

PROGRAM NAME: 2DFFT
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2, CDC Cyber 74
SPECIAL PERIPHERALS: Magnetic Tape Drives
Disk Files

PURPOSE: 2DFFT performs a two dimensional Fast Fourier Transform on image data. Options in the program include:

- a) Forward transform
- b) Inverse transform
- c) Image filtering by specification of one of many optional filters.

PROGRAM NAME: FILTER
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2
SPECIAL PERIPHERALS: Image display system and 1 tape drive
or 2 magnetic tape drives

PURPOSE: FILTER is a program whose concept was taken from a technique used at USGS, Flagstaff, Az. for high frequency enhancement of image data. The basic method involves taking a windowing approach to create a low pass filtered image and subtracting that image from the original image. The resultant is essentially a high pass filtered image. By adding the high pass filtered image to the original image the high frequency enhancement is achieved.

PROGRAM NAME: IMGRID (Harvard U.)
LANGUAGE: FORTRAN IV
COMPUTER: CDC Cyber 74, NOVA 2 (Data General)
SPECIAL PERIPHERALS: Disk Files

PURPOSE: IMGRID is a general purpose geographic data base manipulation program developed by the Harvard University Graduate School of Landscape Design. This program provides for manipulation of multiple data variables related to the same geographic area and in a gridded format. Spatial searching, statistics generation, and modeling via multivariable weighting parameters are key to its analysis capability. Visual and environmental impact analysis are two uses of such a system.

PROGRAM NAME: TOPO
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2
SPECIAL PERIPHERALS: Tape Drive
Image Display

PURPOSE: After breaking down the NCIC topographic tapes into NOVA 4096 record blocks with the CDC Cyber 74, TOPO unpacks and displays the data on the imaging system. Various scales may be represented by selection of parameters in the program.

PROGRAM NAME: FILTER
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2
SPECIAL PERIPHERALS: Image display system and 1 tape drive
or 2 magnetic tape drives

PURPOSE: FILTER is a program whose concept was taken from a technique used at USGS, Flagstaff, Az. for high frequency enhancement of image data. The basic method involves taking a windowing approach to create a low pass filtered image and subtracting that image from the original image. The resultant is essentially a high pass filtered image. By adding the high pass filtered image to the original image the high frequency enhancement is achieved.

PROGRAM NAME: IMGRID (Harvard U.)
LANGUAGE: FORTRAN IV
COMPUTER: CDC Cyber 74, NOVA 2 (Data General)
SPECIAL PERIPHERALS: Disk Files

PURPOSE: IMGRID is a general purpose geographic data base manipulation program developed by the Harvard University Graduate School of Landscape Design. This program provides for manipulation of multiple data variables related to the same geographic area and in a gridded format. Spatial searching, statistics generation, and modeling via multivariable weighting parameters are key to its analysis capability. Visual and environmental impact analysis are two uses of such a system.

PROGRAM NAME: TOPO
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2
SPECIAL PERIPHERALS: Tape Drive
Image Display

PURPOSE: After breaking down the NCIC topographic tapes into NOVA 4096 record blocks with the CDC Cyber 74, TOPO unpacks and displays the data on the imaging system. Various scales may be represented by selection of parameters in the program.

PROGRAM NAME: SCORECARD
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2
SPECIAL PERIPHERALS: Tape Drive

PURPOSE: SCORECARD performs a maximum likelihood classification on specific polygons within a data set. This is used to evaluate accuracy of classification by comparing the classified data for test fields to known ground truth.

PROGRAM NAME: THERMAL
LANGUAGE: FORTRAN
COMPUTER: Data General NOVA 2
SPECIAL PERIPHERALS: Tape Drive
 Image Display

PURPOSE: THERMAL is a program designed to unpack and display digital thermal data from a NASA owned thermal scanner - RS18.

PROGRAM NAME: CHAN24
LANGUAGE: FORTRAN IV
COMPUTER: Data General NOVA 2
SPECIAL PERIPHERALS: Image Display
 Tape Drives

PURPOSE: CHAN24 unpacks data from the Bendix 24 channel aircraft scanner and reformats the data such that single channels may be accessed and written out to either tape or an image display.

PROGRAM NAME: M6OCL
LANGUAGE: FORTRAN IV
COMPUTER(S): Data General NOVA 2
SPECIAL PERIPHERALS: 2 Tape Drives
Disk

PURPOSE: M6OCL uses a maximum likelihood decision rule to classify Landsat data into one of up to sixty classes for which means and covariances are available on a disk file. Thresholds (probability of correct classification) are output for each record of data. This program is a record by record classifier. One record is read from tape, classified, and then written to an output tape before the next record is processed.

PROGRAM NAME: ASTEP
LANGUAGE: FORTRAN IV
COMPUTER: CDC Cyber 74, UNIVAC 1108
SPECIAL PERIPHERALS: Tape Drives
Disk Files

PURPOSE: ASTEP is a general purpose earth resources analysis program developed by TRW Systems for NASA Johnson Space Center. The acronym ASTEP stands for Algorithm Simulation Test and Evaluation Program. The program is of modular construction with standardized input-output such that they are essentially transparent to the user. Different classification, clustering, statistics generating, or feature selection algorithms may be tested against one another with a minimum of programming change to the whole system. In addition to its usefulness as an algorithm test bed, ASTEP has been used effectively as an operational, interactive classification system for Landsat data.

PROGRAM NAME: SEARCH (NASA/ERL - Ronnie Pearson)

LANGUAGE: FORTRAN IV

COMPUTER: SEL 32/55

SPECIAL PERIPHERALS: 1 Tape Drive
Disk Files

PURPOSE/DESCRIPTION: SEARCH is a program developed by Ronnie Pearson of NASA/ERL for unsupervised development of signatures for use in a maximum likelihood classification scheme. A 3 x 3 or 6 x 6 pixel moving window is used in a single iteration through the raw Landsat data to form candidate signatures. A maximum number of acceptable signatures is specified and a divergence criterion is used for merging, splitting, and selection of signatures. This program normally takes approximately one hour for development of signatures for one Landsat scene. Auxiliary programs are available for intuitively assigning color values for each class for use on a color display based on a two dimensional plot of the signature means for channels 2 and 4 of Landsat MSS data.

PROGRAM NAME: COORD

LANGUAGE: FORTRAN IV

COMPUTER(S): CDC Cyber 74

SPECIAL PERIPHERALS: Disk Files

PURPOSE: COORD accepts pairs of Latitude-Longitude or UTM coordinates and Landsat pixel coordinates for Ground Control Points(GCP) and computes a least squares fit of the transformation matrix needed to map Landsat data into a standard coordinate system.

PROGRAM NAME: CLUSTER
LANGUAGE: FORTRAN IV
COMPUTER(S): Data General NOVA 2, CDC Cyber 74
SPECIAL PERIPHERALS: 2 Tape Drives

PURPOSE: CLUSTER is a sequential clustering algorithm which creates an unsupervised classification of Landsat or aircraft multi-spectral scanner (MSS) data using a Euclidean distance measure as a decision criterion. This system decides how many "different" types of land cover there are in a MSS scene. This system is dependent on user input parameters which specify the criteria for number of clusters, merging, creation of new clusters, and exclusion of clusters. This technique is often used to define training fields for supervised classification.

PROGRAM NAME: CLUST (NASA/ERL - Ronnie Pearson)
LANGUAGE: FORTRAN IV
COMPUTER: SEL 32/55
SPECIAL PERIPHERALS: 1 Tape Drive
Disk Files

PURPOSE/DESCRIPTION: CLUST is structurally similar to the SEARCH program for unsupervised development of class signatures. CLUST, however, uses a Euclidean distance measure in its sort and merge control logic for clusters. CLUST is often more useful than SEARCH in very broken terrain where fields of 40 acres are not common.

PROGRAM NAME: SUPERG (NASA/ERL - Marcellus Graham)

LANGUAGE: FORTRAN IV

COMPUTER: SEL 32/55

SPECIAL PERIPHERALS: 2 Tape Drives
Disk Files

PURPOSE/DESCRIPTION: SUPERG is a rectification program for Landsat MSS data which includes 2nd order mirror corrections into a least squares determination of a covariance matrix for conversion of Landsat pixels into a UTM coordinate system. The data are resampled along scan lines to satisfy scaling considerations and the output file contains new pixels which are directly related to the UTM system. Between 10 and 30 Ground Control Points are suggested for complete determination of the transformation. Rotation of the data to true North is not accomplished by this program.

PROGRAM NAME: TRAIN

LANGUAGE: FORTRAN IV

COMPUTER: Data General NOVA II

SPECIAL PERIPHERALS: COMTAL Interactive Color Video Display (3 images)
Magnetic Tape Drive
Disk Files

PURPOSE: TRAIN is an interactive training field selection and statistics generation program for Landsat digital analysis. In conjunction with a video display system with a cursor or joystick, a subset of a Landsat image may be selected by drawing an arbitrarily shaped polygon (up to 100 vertices) around an area on the display screen. The program keeps track of position on the input Goddard format CCT and calculates the normal statistics (mean and covariance) of the selected training fields. Histograms of the multivariate distributions are displayed on the display screen and the mean and variance, polygon vertices, and histograms may be saved on a disk file. The program acts in a question-answer mode which requires no knowledge of computer languages by the user.

A parallelopiped classifier is also implemented in this program which will classify a 256 x 256 element scene for one class in near real time.

APPENDIX B
PROGRAM LISTINGS

```

C*****
C
C      ADDSIGM
C
C      THIS ROUTINE ADDS SPECIFIED SIGNATURES
C      TO FORM NEW MEAN AND COVARIANCE
C
C      SEQUENCE:  ADDSIGM INPUT SIGFIL NEWSIG
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION COV1(4,4),COV2(4,4),NUM1(12),NUM2(12),IORDER(60)
C      DIMENSION ID(6),AMEAN(4,60),BCOV(4,4,60),I1(30),I2(30),I3(30)
C      DIMENSION NP(60),NAM1(60),NAM2(60),NAM3(60),NAM4(60)
C      DIMENSION VM1(12),VM2(12),VM(4),ISW(2)
C      COMMON/NPP/NP
C      ND=4
C      IP=12
C
C      COMARG AND OPEN STATEMENTS
C
C      CALL OPEN(1,"COM.CM",1,IERR)
C      CALL COMARG(1,I1,ISW,IERR)
C      CALL COMARG(1,I1,ISW,IERR)
C      CALL COMARG(1,I2,ISW,IERR)
C      CALL COMARG(1,I3,ISW,IERR)
C      CALL OPEN(2,I1,0,IE)
C      CALL FOPEN(4,I3,"B")
C
C      TYPE "  INPUT NEWSIG NAME  "
C      READ(11,200)N1,N2,N3,N4
C      READ(2)NSIG
C      WRITE(IP)NSIG
C
C      GET SIGNATURES LISTED IN INPUT FILE
C
C      CALL GSIG(AMEAN,BCOV,NSIG,2,3,12,IORDER,NAM1,NAM2,NAM3,NAM4)
C      NUM1(1)=NP(1)
C      DO (J=1,4)
C      :   VM1(J)=AMEAN(J,1)
C      :   DO (L=1,4)
C      :   :   COV1(J,L)=BCOV(J,L,1)
C      :   :   ...FIN
C      :   ...FIN
C      NS1=NSIG-1

```

C
C
C

LOOP TO COMBINE NSIG SIGNATURES

C
C
C
C

```

DO (K=1, NS1)
:   KP1=K+1
:   DO (J=1, 4)
:     :   VM2(J)=AMEAN(J, KP1)
:     :   DO (L=1, 4)
:     :     :   COV2(J, L)=BCOV(J, L, KP1)
:     :     :   ...FIN
:     :   ...FIN
:   NUM2(1)=NP(KP1)
:
:   CALL ADDSIG(COV1, COV2, VM1, VM2, ND, NUM1, NUM2)
:
: ...FIN
DO (I=1, 4)
:   VM(I)=VM1(I)
: ...FIN
WRITE(IP, 201) N1, N2, N3, N4
WRITE(IP) NUM1(1), VM, COV1
DO (L=1, 6) ID(L)=0
WRITE BINARY(4) ID
WRITE BINARY(4) N1, N2, N3, N4
WRITE BINARY(4) NUM1(1), VM, COV1
200 FORMAT(4A2)
201 FORMAT(2X, 4A2)
STOP
END

```

```

C*****
C
C      ALARM      (SUBROUTINE)
C
C      THIS SUBROUTINE ALARMS 1 CLASS ON THE COMTAL
C      CORRESPONDING TO THE LAST SIGNATURE,
C      IT ALARMS TO GRAPHICS OVERLAY #IOV
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMERS:  NICKOLAS L. FAUST
C                   ROBERT A. MADDOX
C*****
SUBROUTINE ALRM2( IOV)
  DIMENSION IMIN(4), IMAX(4), IX2(0:255,3)
  COMMON/DTRANS/IMAG1(0:255), IXD(0:511,4), IMAG(0:511)
  COMMON/HISTX/ICOUNT(4,100)
  EQUIVALENCE (IXD,IX2)
  DO (I=0,15) IMAG(I)=0
  DO (I=0,255) CALL GWR( IOV, I, IMAG, 16)
  DO (I=1,4)
    : DO (J=1,100)
    :   IF( ICOUNT(I,J).GT.0) GO TO 30
    :   :..FIN
30  : CONTINUE
    :   IMIN(I)=J
    :   DO (JJ=1,100)
    :     : N=101-JJ
    :     : IF( ICOUNT(I,N).GT.0) GO TO 50
    :     :..FIN
50  : CONTINUE
    :   IMAX(I)=N
    :..FIN
  IMIN1=IMIN(1)
  IMIN2=IMIN(2)
  IMIN4=IMIN(4)
  IMAX1=IMAX(1)
  IMAX2=IMAX(2)
  IMAX4=IMAX(4)
  DO (K=0,255)
    : CALL IMRD(0,K,IMAG(0),128)
    : CALL IMRD(1,K,IMAG(128),128)
    : CALL IMRD(2,K,IMAG(256),128)
    : CALL UPAC8( IMAG, IX2(0,1), 384)
    : DO (L=0,255)
    :   : IMAG1(L)=0
    :   : IF (IX2(L,1).LT.IMIN1) GOTO 70
    :   : IF (IX2(L,1).GT.IMAX1) GOTO 70
    :   : IF (IX2(L,2).LT.IMIN2) GOTO 70
    :   : IF (IX2(L,2).GT.IMAX2) GOTO 70
    :   : IF (IX2(L,3).LT.IMIN4) GOTO 70
    :   : IF (IX2(L,3).GT.IMAX4) GOTO 70
    :   : IMAG1(L)=1
70  :   : CONTINUE
    :   :..FIN
    : CALL PACB( IMAG1, IMAG, 16)
    : CALL GWR( IOV, K, IMAG, 16)
    :..FIN
  RETURN
END

```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```

C*****
C
C      ANCDIS   (SUBROUTINE)
C
C      PURPOSE:
C      COMPUTES THE DISTANCES AND ANGLES BETWEEN A SET OF
C      VECTORS OF ARBITRARY DIMENSION
C
C      DESCRIPTION OF PARAMETERS
C
C      INPUT:
C      CALLING SEQUENCE
C      VM - NVM VECTORS OF DIMENSION ND , STORED BY COLUMNS
C      NVM - NUMBER OF VECTORS
C      ND - DIMENSION OF EACH VECTOR
C      IDISF - .EQ.1 COMPUTE EUCLIDEAN DISTANCE
C              .NE.1 COMPUTE L1 DISTANCE
C
C      OUTPUT
C      CALLING SEQUENCE
C      R - AN NVM BY NVM MATRIX WITH I TH - J TH ELEMENT
C      CORRESPONDING TO I TH - J TH VECTORS AND EQUALS ANGL
C      IF ABOVE DIAGONAL AND EQUALS DISTANCE IF BELOW
C      DIAGONAL
C
C*****
C      CREATED AT NASA/JSC (ASTEP)
C*****
C      SUBROUTINE ANCDIS(VM,NVM,ND,IDISF,R)
C      DIMENSION VM(ND,NVM),R(NVM,NVM)
C      N = NVM - 1
C      DO 20 J = 1,N
C      R(J,J) = 0.
C      I1 = J + 1
C      DO 10 I = I1,NVM
C      IF(IDISF.EQ.1) GO TO 6
C      D=0.
C      DO 4 K=1,ND
C      4 D=D+ABS(VM(K,J)-VM(K,I))
C      GO TO 8
C      6 CALL EDIST(VM(1,J),VM(1,I),ND,D)
C      8 CONTINUE
C      CALL ANGLE(VM(1,J),VM(1,I),ND,A)
C      R(I,J) = D
C      R(J,I) = A
C      10 CONTINUE
C      20 CONTINUE
C      R(NVM,NVM) = 0.
C      RETURN
C      END

```


ANGLE (SUBROUTINE)

PURPOSE:

COMPUTES THE ANGLE BETWEEN TWO VECTORS OF ARBITRARY
DIMENSION

DESCRIPTION OF PARAMETERS

INPUT

CALLING SEQUENCE

V1 - 1 ST VECTOR

V2 - 2 ND VECTOR

ND - DIMENSION OF V1 AND V2

OUTPUT

CALLING SEQUENCE

A - ANGLE BETWEEN V1 AND V2 IN DEGREES

CREATED AT NASA/JSC (ASTEP)

SUBROUTINE ANGLE (V1, V2, ND, A)

DIMENSION V1(ND),V2(ND)

D1 = 0.

D2 = 0.

A = 0.

DO 10 I = 1,ND

D1 = D1 + V1(I)**2

D2 = D2 + V2(I)**2

10 A = A + V1(I)*V2(I)

IF (D1.EQ.0.0 .OR. D2.EQ.0.0) GO TO 20

A = A/(SQRT(D1)*SQRT(D2))

A = 57.29578*ACOS(A)

15 RETURN

20 A = 90.0

GO TO 15

END

ORIGINAL PAGE IS
OF POOR QUALITY

AUTO811Z

DESIGNED TO OUTPUT CLASSIFIED TAPE IN PATTERN FORMAT
TO DISPLAY ON THE VERSATEC

SEQUENCE: AUTO811Z MTU:F INPUTFILE

CREATED AT GEORGIA TECH EES

PROGRAMMER: MICHAEL D. FURMAN

```
INTEGER IMAG(180), ITAPE(10), IOUT(360), IORDER(25,15), ISW(2)
INTEGER NC(25), IWORK(4100), IMAG3(180), INEM(0:255), IMAG2(360)
INTEGER IWORK1(360), IWORK2(360), IWORK3(360), IWORK4(360)
INTEGER IWORK5(360), IWORK6(360), IWORK7(360), IWORK8(360)
INTEGER IWORK9(360), IWORK10(360), IWORK11(360), IWORK12(360)
INTEGER IMES(15,40), IFLD(10)
EQUIVALENCE (IWORK(1), IWORK1(1))
EQUIVALENCE (IWORK(359), IWORK2(1))
EQUIVALENCE (IWORK(717), IWORK3(1))
EQUIVALENCE (IWORK(1075), IWORK4(1))
EQUIVALENCE (IWORK(1433), IWORK5(1))
EQUIVALENCE (IWORK(1791), IWORK6(1))
EQUIVALENCE (IWORK(2149), IWORK7(1))
EQUIVALENCE (IWORK(2507), IWORK8(1))
EQUIVALENCE (IWORK(2865), IWORK9(1))
EQUIVALENCE (IWORK(3223), IWORK10(1))
EQUIVALENCE (IWORK(3581), IWORK11(1))
EQUIVALENCE (IWORK(3939), IWORK12(1))
COMMON/DUM/ID(165), IWORK
DATA ID/
10,0,0,0,0,0,0,0,0,0,0,0,
20,176K,162K,102K,102K,102K,102K,102K,102K,176K,0,
30,0,176K,176K,176K,176K,176K,176K,176K,176K,0,0,
4300K,340K,160K,160K,60K,30K,14K,16K,16K,7,3,
5176K,74K,30K,0,201K,303K,201K,0,30K,74K,176K,
6303K,303K,303K,303K,0,0,0,303K,303K,303K,303K,
730K,30K,30K,30K,30K,30K,30K,30K,30K,30K,30K,
830K,30K,30K,30K,377K,377K,377K,30K,30K,30K,30K,
93,7,16K,16K,14K,30K,60K,160K,160K,340K,300K,
1377K,301K,241K,231K,231K,231K,231K,231K,205K,203K,377K,
1201K,303K,347K,176K,74K,30K,74K,176K,347K,303K,201K,
2374K,370K,361K,361K,363K,347K,317K,217K,217K,37K,77K,
377K,37K,217K,217K,317K,347K,363K,361K,361K,370K,374K,
4377K,377K,377K,303K,303K,303K,303K,303K,377K,377K,377K,
5377K,377K,377K,377K,377K,377K,377K,377K,377K,377K,
CALL OPEN(1,"COM.CM",1,IERR)
CALL COMARG(1,ITAPE,ISW,IERR)
CALL COMARG(1,ITAPE,ISW,IERR)
CALL COMARG(1,IFLD,ISW,IERR)
CALL FOPEN(3,IFLD,"B")
```

```

IO=3
CALL NTOPD(2, ITAPE, 0, IE)
DO (I1=1, 25)
: DO (I2=1, 15) IORDER(I1, I2)=0
...FIN
TYPE "DOES INPUT TAPE HAVE THRESHOLD? (2=YES, 1=NO) "
READ(IO) ITH
TYPE "LINE"
READ(IO) ILINE
TYPE "MAP PORTION (1 - 12) ", IANS2
READ(IO) IANS2
TYPE "LENGTH OF DATA "
READ(IO) IST
IEND=IEL+359
IF (IEND.GT.4100) IEND=4100
TYPE "PRODUCE OVERALL MAP? (1=YES) "
READ(IO) IANS
TYPE "NUMBER OF GENERALIZED CLASSES "
READ(IO) IGEN
DO (I=1, IGEN)
: TYPE "INPUT CLASS DESCRIPTION "
: READ(IO, 101) (INES(I, K), K=1, 20)
: TYPE "NUMBER OF SUBCLASSES "
: READ(IO) NC(I)
: NC1=NC(I)
: TYPE "INPUT SUBCLASSES "
: READ(IO) (IORDER(I, K), K=1, NC1)
: IF (I.NE.IGEN) TYPE "NEXT GROUP"
...FIN
50 DO (I=1, 253) IMAC2(I)=0
DO (I=1, 180) IMAC3(I)=-1
DO (I=0, 255) INEM(I)=0
WRITE(12)
L2=154
DO (I=1, ILINE) CALL MTDIO(2, 0, IWORK2, IS, IE, IC)
IF (IANS.EQ.1)
: DO (I=1, 180) IOUT(I)=0
: DO (M5=1, IGEN)
: : L3=NC(M5)
: : DO (J2=1, L3) INEM(IORDER(M5, J2))=L2
: : WRITE(12, 100) (INES(M5, K), K=1, 20) (IORDER(M5, I), I=1, L3)
: : DO (J=1, 11)
: : : IOUT(15)=ID(L2+J)
: : : CALL MTX(IOUT, 180)
: : : ...FIN
: : L2=L2-11
: : ...FIN
: WRITE(12)
...FIN

```

```

DO (M1=1, IGEN)
: IF ( IANS.NE.1)
: : L3=NC(M1)
: : WRITE(12,100) ( IMES(M1,K), K=1,20) ( IORDER(M1,I), I=1,L3)
: : DO (I=1,180) IOUT(I)=0
: : DO (I=0,255) IMEM(I)=0
: : DO (J=1,11)
: : : L2=154
: : : DO (J2=1,L3)
: : : : IMEM(IORDER(M1,J2))=L2
: : : : IOUT(J2*4+9)=ID(L2+J)
: : : : L2=L2-11
: : : : ...FIN
: : : CALL MTX(IOUT,180)
: : : ...FIN
: : WRITE(12)
: : ...FIN
: DO (I=1,8) CALL MTX(IMAG3,180)
: DO (M2=1,IST)
: : DO (I=1,ITH)
: : : CALL MIDIO(2,0,IWORK,IS,IE,IC)
: : : IF (IC.LT.5) GOTO 55
: : : ...FIN
: : CONDITIONAL
: : : ( IANS2.EQ.1) CALL AUTECC(IWORK1,IMEM,11)
: : : ( IANS2.EQ.2) CALL AUTECC(IWORK2,IMEM,11)
: : : ( IANS2.EQ.3) CALL AUTECC(IWORK3,IMEM,11)
: : : ( IANS2.EQ.4) CALL AUTECC(IWORK4,IMEM,11)
: : : ( IANS2.EQ.5) CALL AUTECC(IWORK5,IMEM,11)
: : : ( IANS2.EQ.6) CALL AUTECC(IWORK6,IMEM,11)
: : : ( IANS2.EQ.7) CALL AUTECC(IWORK7,IMEM,11)
: : : ( IANS2.EQ.8) CALL AUTECC(IWORK8,IMEM,11)
: : : ( IANS2.EQ.9) CALL AUTECC(IWORK9,IMEM,11)
: : : ( IANS2.EQ.10) CALL AUTECC(IWORK10,IMEM,11)
: : : ( IANS2.EQ.11) CALL AUTECC(IWORK11,IMEM,11)
: : : ( IANS2.EQ.12) CALL AUTECC(IWORK12,IMEM,11)
: : : ...FIN
: : ...FIN
55 : DO (I=1,11) CALL MTX(IMAG3,180)
: DO (I=1,700)
: : DO (J=1,360) IOUT(J)=ID(I)
: : DO (N10=1,1100) N11=N10
: : : CALL PAC8(IOUT,IMAG,360)
: : : CALL MTX(IOUT,180)
: : : ...FIN
: : CALL MIDIO(2,10000K,IWORK,IS,IE,IC)
: : DO (I=1,ILINE) CALL MIDIO(2,0,IWORK2,IS,IE,IC)
: : IF ( IANS.EQ.1)
: : : IANS=0
: : : GOTO 50
: : : ...FIN
: : ...FIN
: DO (I=1,10) WRITE(12)
STOP FINISHED
100 FORMAT(1X,/,1X,20A1,/, " CLASSES ",20I4)
101 FORMAT(20A1)
END

```

```

C*****
C
C      CHAN24
C
C      UNPACKS AND REFORMATS BENDIX 24-CHANNEL DATA FOR DISPLAY
C
C      SEQUENCE CHAN24 MTU:F MTU:F
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  MICHAEL D. FURMAN
C*****
C      INTEGER IDATA(4100), IMAG(512), IMAG2(256), ITAPE1(5), ITAPE2(5)
C      INTEGER IWORK(256), IHEAD(30), IX(3), IY(3), ISW(2)
C      COMMON IX,IY
C      EQUIVALENCE (IX(1),IX1),(IX(2),IX2),(IY(1),IY1),(IY(3),IY3)
C      CALL OPEN(1,"COM.CM",1,IERR)
C      CALL COMARG(1,ITAPE1,ISW,IERR)
C      CALL COMARG(1,ITAPE1,ISW,IERR)
C      CALL COMARG(1,ITAPE2,ISW,IERR)
C      DO (I=1,30) IHEAD(I)=0
C
C 10  FORMAT(1X,I2,"/",I2,"/",I2)
C 11  FORMAT(1X,"INPUT TAPE NO.",Z)
C 12  FORMAT(1X,10A2)
C 13  FORMAT(1X,"BLOCK SIZE= ",I3)
C 14  FORMAT(10A2)
C
C      TYPE "DATE M,D,Y"
C      ACCEPT IHEAD(3),IHEAD(4),IHEAD(5)
C      TYPE "INPUT TAPE NUMBER? XXXX"
C      ACCEPT IHEAD(1)
C      TYPE "COMMENTS ON RUN -- 20 CHARACTERS"
C      READ(11,14) (IHEAD(I),I=10,20)
C      CALL COLORSUB
C      TYPE "FAST SCAN? (2=YES, 1=NO) "
C      ACCEPT IFSCAN
C      TYPE "DATA BLOCKS TO SKIP?"
C      ACCEPT ISKB
C      TYPE "INPUT BLOW-UP FACTOR"
C      ACCEPT IBLUP
C      IBLUPM1=IBLUP-1

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
IELEM=1
IF (IFSCAN.EQ.1)
: TYPE "START WITH ELEMENT?"
: ACCEPT IELEM
...FIN
CALL SCALESUB
TYPE "INPUT CHANNEL NUMBER(1-24) "
ACCEPT ICHAN
ISKB=(ICHAN+2)/3+(ISKB*9)+2
INUM=5+ICHAN-(((ICHAN+2)/3+1)*3)
L=INUM*393+2+IELEM
LAST=L+256+((IFSCAN-1)*92)
IF (IELEM.GT.94) LAST=351+(INUM*393)
CALL MTOPD(3,ITAPE1,0,IER)
CALL MTOPD(4,ITAPE2,0,IER)
DO (I=1,ISKB) CALL MTDIO(3,0,IDATA,IS,IER,ICNT)
M2=512/IBLUP
M3=0
DO (M1=1,M2)
: DO (N=1,IFSCAN)
: : CALL MTDIO(3,30010K,IDATA,IS,IE,IC)
: : CALL MTDIO(3,0,IDATA,IS,IE,IC)
: : IF (IC.LT.10) GOTO 150
: ...FIN
: J=1
: DO (K=L, LAST, IFSCAN)
: : IMAG(J)=IDATA(K)
: : J=J+1
: ...FIN
: BLOWUP-AREA
: DO (I=1,IBLUP)
: : CALL IMWRITE(0,M3,IMAG2,256)
: : M3=M3+1
: ...FIN
...FIN
150 DO (L1=1,3)
: PAUSE -- POSITION CURSER AND HIT RETURN
: CALL RTARG(IX(L1),IY(L1))
...FIN
IX1=IX1/2
IX2=IX2/2
DO (I=1,3,2)
: CALL IMREAD(0,IY(I),IWORK,256)
: DO (I2=IX1,IX2) IWORK(I2)=255
: CALL IMWRITE(0,IY(I),IWORK,256)
...FIN
DO (I=IY1,IY3)
: CALL IMREAD(0,I,IWORK,256)
: IWORK(IX1)=255
: IWORK(IX2)=255
: CALL IMWRITE(0,I,IWORK,256)
...FIN
```

```

TYPE "PROPER RECTANGLE? (1=YES; 2=NO) "
ACCEPT IANS
IF (IANS.EQ.2) GOTO 150
CALL MTDIO(3,10000K, IDATA, IS, IE, ICNT)
IX1=IX1*IFSCAN+IELEM-1
IX2=IX2*IFSCAN+IELEM-1
IY1=IY1*IFSCAN
IY3=IY3*IFSCAN
IPAS=9*IY1
IDIF=(IY3-IY1)
IDIF2=IX2-IX1
WRITE-HEADER
CALL MTDIO(3,30001K+IPAS, IDATA, IS, IE, ICNT)
DO (I=1, IDIF)
: CALL MTDIO(3,30001K, IDATA, IS, IE, ICNT)
: DO (K5=1,8)
: : CALL MTDIO(3,0, IDATA, IS, IE, ICNT)
: : DO (L6=3,789,393)
: : : J=1
: : : DO (L5=IX1,IX2)
: : : : IWORK(J)=IDATA(L5+L6)
: : : : J=J+1
: : : ...FIN
: : : CALL MTDIO(4,50000K+IDIF2, IWORK, IS, IE, ICNT)
: : : ...FIN
: : ...FIN
: ...FIN
DO (I=1,10) CALL MTDIO(4,60000K, IWORK, IS, IE, ICNT)
CALL MTDIO(4,10000K, IWORK, IS, IE, ICNT)
CALL MTDIO(3,10000K, IDATA, IS, IE, ICNT)
STOP

```

C

```

TO WRITE-HEADER
: WRITE(12,11)
: WRITE(12) IHEAD(1)
: WRITE(12,10) IHEAD(3), IHEAD(4), IHEAD(5)
: WRITE(12,12) (IHEAD(I), I=10,20)
: WRITE(12,13) IDIF2
: CALL MTDIO(4,50036K, IHEAD, IS, IE, ICNT)
...FIN

```

C

```

TO BLOWUP-AREA
: DO (K7=0,255)
: : K9=256-K7
: : KDB=K9/IBLUP
: : IMAG(K9)=IMAG(KDB)
: : ...FIN
: CALL PAC8(IMAG, IMAG2,512)
...FIN

```

200 STOP FIELD ERROR -- CHAN24 MTU:F MTU:F
END

```

C*****
C
C      CLASIFY
C
C      THIS ROUTINE CLASSIFIES POLYGONS WITH A MAXLIK CLASSIFIER
C      A TABLE LOOK-UP FEATURE IS INCLUDED
C
C      SEQUENCE:  CLASIFY INPUT DATAP MAXTAP SIGFIL
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMERS:  NICKOLAS L. FAUST
C                   ROBERT A. MADDOX
C*****
C      DIMENSION AMEAN(4,60),BCOV(4,4,60),IVX(101),IVY(101),DET(60)
C      1,LA(133),COUNT(0:61),IBUF(1640),CLCON(61)
C      2,NAM1(60),NAM2(60),NAM3(60),NAM4(60),B1(4,4),B2(4,4)
C      DIMENSION JBUF(4,810),FIELD(17),ISWS(2),IORD(0:61)
C      DIMENSION DUM(61),DUM1(4,4,60),DUM2(4,60),ITABL(5,300)
C      COMMON JBUF
C      EQUIVALENCE (JBUF,DUM), (JBUF,DUM1), (JBUF,DUM2), (JBUF(1,20),DET),
C      1(JBUF,B1), (JBUF(1,10),B2)
C
C      COMARG CALLS  AND OPEN STATENENTS
C
C      CALL OPEN(1,"COM.CM",1,IE)
C      CALL COMARG(1,FIELD,ISWS,IE)      ;GET PROGRAM NAME
C      CALL COMARG(1,FIELD,ISWS,IE)      ;GET INPUT FILE NAME
C      CALL OPEN(2,FIELD,0,IE)           ;OPEN INPUT FILE
C      CALL COMARG(1,FIELD,ISWS,IE)      ;GET DATA TAPE NAME
C      CALL MTOPD(3,FIELD,0,IE)          ;OPEN TAPE FILE
C      CALL COMARG(1,FIELD,ISWS,IE)      ;GET OUTPUT TAPE NAME
C      CALL MTOPD(5,FIELD,0,IE)          ;OPEN TAPE OUTPUT FILE
C      CALL COMARG(1,FIELD,ISWS,IE)      ;GET SIGNATURE FILE NAME
C
C      IP=12
C      ND=4
C      BNV=-1.E+12      ;BIG NEGATIVE NUMBER
C      ITPTR=0          ;INITIAL TABLE POINTER
C      READ(2,200) NSIG
C
C      GET SIGNATURES  FOR CLASSIFICATION
C
C      CALL GSIG(AMEAN,BCOV,NSIG,FIELD,NAM1,NAM2,NAM3,NAM4,CLCON)
C      DO (L=1,NSIG)
C      :   DO (JH=1,4)
C      :   :   DO (JI=1,4)
C      :   :   :   B2(JH,JI)=BCOV(JH,JI,L)
C      :   :   :   ...FIN
C      :   :   ...FIN
C      :   CALL SYMINV(B2,B1,DETP,4)
C      :   DO (JH=1,4)
C      :   :   DO (JI=1,4)
C      :   :   :   BCOV(JH,JI,L)=B1(JH,JI)
C      :   :   :   ...FIN
C      :   :   ...FIN
C      :   DIVIDE DIAGONAL ENTRIES OF COVARIANCE MATRIX BY TWO
C      :   DO (J=1,4) BCOV(J,J,L)=BCOV(J,J,L)/2.
C      :   GET NATURAL LOG OF DETERMINANT
C      :   DET(L)=ALOG(DETP)
C      :   ...FIN

```



```

C
IYMIN=10000
IXMIN=10000
IYMAX=0
WRITE(IP,202)
READ(2,200) NC
DO (I=1,NC)
: WRITE(IP,205) I
: READ(2,206) IVX(I), IVY(I)
: WRITE(IP,206) IVX(I), IVY(I)
: IF(IVY(I).LT.IYMIN) IYMIN=IVY(I)
: IF(IVX(I).LT.IXMIN) IXMIN=IVX(I)
: IF(IVY(I).GT.IYMAX) IYMAX=IVY(I)
...FIN
IVX(NC+1)=IVX(1)
IVY(NC+1)=IVY(1)
NV=NC
IDEL=IYMAX-IYMIN
ISKIP=IYNIN+1

```

```

C
C
C
C
C
      SET UP THE APRIORI INFORMATION EITHER FROM,
      A. THE INPUT FILE IF 'IAPR' EQUALS 1 IN INPUT FILE
      OR   B. SET APRIORI FOR ALL CLASSES EQUAL TO 1.0

```

```

C
C
C
C
C
READ(2,201) NTE           ;NUMBER OF TABLE ENTRIES

READ(2,200) ITH
IF(ITH.NE.0)
: READ(2,310) THRS
: THRS=ALOG(THRS)
...FIN
READ(2,200) IAPR
IF(IAPR.EQ.0)
: DO(I=1,NSIG) CLCON(I)=1.
...FIN
WRITE(IP,410)
DO(I=1,NSIG)
: WRITE(IP,420) I,NAM1(I),NAM2(I),NAM3(I),NAM4(I),CLCON(I)
...FIN
WRITE(IP,400)

```

```

C
C
C
      COMPUTE CLASS CONSTANTS

```

```

PILOG=2.*ALOG(2.*3.141593)
DO(I=1,NSIG) CLCON(I)=ALOG(CLCON(I))-.5*DET(I)-PILOG
CLCON(NSIG+1) = BN

```

```

C
C
C
C
      SORT CLASS CONSTANTS FROM LARGEST TO SMALLEST
      AND ALSO REORDER STATISTICS

```

```

DO(J=1,NSIG)
: BG = -1.E+10
: DO(I=1,NSIG)
: : IF(CLCON(I).GT.BG)
: : : BG=CLCON(I)
: : : IND=I
: : ...FIN
: ...FIN
: IORD(J)=IND
: DUM(J)=CLCON(IND)
: CLCON(IND)=BN
...FIN
IORD(61)=61
IORD(0)=0

```

C
C
C

REORDER CLASS CONSTANT ARRAY

```
DO (J=1,NSIG) CLCON(J)=DUM(J)
CLCON(NSIG+1)=BNV
```

ORIGINAL PAGE IS
OF POOR QUALITYC
C
C

REORDER COVARIANCE MATRICES

```
DO (I=1,NSIG)
: DO (J=1,ND)
: : DO (K=1,ND) DUM1(J,K,I) = BCOV(J,K,IORD(I))
: : ...FIN
: ...FIN
DO (I=1,NSIG)
: DO (J=1,ND)
: : DO (K=1,ND) BCOV(J,K,I)=DUM1(J,K,I)
: : ...FIN
: ...FIN
```

C
C
C

REORDER MEAN VECTORS

```
DO (I=1,NSIG)
: DO (J=1,ND) DUM2(J,I)=AMEAN(J,IORD(I))
: ...FIN
DO (I=1,NSIG)
: DO (J=1,ND) AMEAN(J,I)=DUM2(J,I)
: ...FIN
```

C
C
C

SKIP RECORDS

```
CALL MTDIO(3,30000K+ISKIP,IBUF,IS,IEE)
DO (LL=1,812) IBUF(LL)=0
IBUF(1)=IDEL
IBUF(2)=IXMIN
CALL MTDIO(5,50000K+810,IBUF,IS,IER,NW)
ISW=2
JQ=810
TOT=0
TOTHTS=0
DO (I=0,61) COUNT(I)=0
DO (I=1,5)
: DO (J=1,NTE)
: : ITABL(I,J)=0
: : ...FIN
: ...FIN
CALL FGTIME(IHR,IMIN,ISEC)
TYPE "LINE PROCESSING BEGAN @",IHR,IMIN,ISEC
DO (J=1,IDEL)
: HTS=0
: CALL MTDIO(3,0,IBUF,IS,IE,NW)
: CALL POLY2(ISKIP+J-1,ISW,IVX,IVY,NV,LA)
: JD=0
: DO (KO=1,JQ)
: : DO (KP=1,4)
: : : JBUF(KP,KO)=0
: : : ...FIN
: : ...FIN
: LA1=LA(1)
: DO (K1=1,LA1)
: : LSUB=2*K1
: : L2=LSUB+1
: : JS=LA(LSUB)
: : JF=LA(L2)
: : JD=JD+JF-JS+1
: : IF(JF.GT.JQ) JF=JQ
: : DO (K2=1,4)
: : : LL=((JS-1)/2)*4+K2
: : : LAST=LL+(JF-JS+1)*2
: : : J1=1
: : : DO (I1=LL,LAST,4)
: : : : JBUF(K2,J1)=ISHFT(IBUF(I1),-8)
: : : : JBUF(K2,J1+1)=IAND(IBUF(I1),377K)
: : : : J1=J1+2
: : : : ...FIN
: : : ...FIN
: : ...FIN
: ...FIN
```

C
C
C

CLASSIFY THE POINTS BY MAXIMUM LIKLIHOOD ALGORITHM

```

: DO (JPT=1,JQ)
:   IF (JBUF(1,JPT).EQ.0) GOTO 1000
:   JB1=JBUF(1,JPT)
:   JB2=JBUF(2,JPT)
:   JB3=JBUF(3,JPT)
:   JB4=JBUF(4,JPT)
:   DO 1010 IC=1,NTE
:   : IF (JB3.NE.ITABL(3,IC)) GOTO 1010
:   : IF (JB2.NE.ITABL(2,IC)) GOTO 1010
:   : IF (JB4.NE.ITABL(4,IC)) GOTO 1010
:   : IF (JB1.NE.ITABL(1,IC)) GOTO 1010
:   : INDEX=ITABL(5,IC)
:   : HITS=HITS+1
:   : GOTO 1050
1010 :   CONTINUE
:   M=1
:   PROB=BNV
1040 :   VD1=JB1-AMEAN(1,M)
:   VD2=JB2-AMEAN(2,M)
:   VD3=JB3-AMEAN(3,M)
:   VD4=JB4-AMEAN(4,M)
:   CLTHR=CLCON(M) - (VD1*VB1*BCOV(1,1,M)
1:   : +VD2*(VD1*BCOV(2,1,M)+VD2*BCOV(2,2,M))
2:   : +VD3*(VD1*BCOV(3,1,M)+VD2*BCOV(3,2,M)
:   : +VD3*BCOV(3,3,M))
3:   : +VD4*(VD1*BCOV(4,1,M)+VD2*BCOV(4,2,M)
4:   : +VD3*BCOV(4,3,M)+VD4*BCOV(4,4,M))
:   IF (CLTHR.LT.PROB) GOTO 1020
:   PROB=CLTHR
:   INDEX=M
1020 :   M=M+1
:   IF (PROB.LT.CLCON(M)) GOTO 1040
:   IF (ITH.EQ.0) GOTO 1030
:   IF (PROB.LT.THRS) INDEX=61
1030 :   ITPTR=ITPTR+1
:   IF (ITPTR.GT.NTE) ITPTR=1
:   ITABL(1,ITPTR)=JB1
:   ITABL(2,ITPTR)=JB2
:   ITABL(3,ITPTR)=JB3
:   ITABL(4,ITPTR)=JB4
:   ITABL(5,ITPTR)=INDEX
:   GOTO 1050
1000 :   INDEX=0
1050 :   IBUF(JPT)=IORD(INDEX)
:   ...FIN
C
C
:   TOT=TOT+JD
:   TOTHITS=TOTHITS+HITS
:   DO (K=1,JQ)
:   :   IBK=IBUF(K)
:   :   COUNT(IBK)=COUNT(IBK)+1
:   :   ...FIN
:   CALL FGTIME(IHR,IMIN,ISEC)
:   WRITE(10,207)J,HITS
:   TYPE "TIME = ",IHR,IMIN,ISEC
:   CALL MTDIO(5,50000K+810,IBUF,IS,IER)
:   ...FIN

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
WRITE(IP,400)
WRITE(IP,430)
DO (K=1,NSIC)
: PER=(COUNT(K)*100)/TOT
: IKK=IORD(K)
: WRITE(IP,440) K,NAM1(K),NAM2(K),NAM3(K),NAM4(K),IKK
1: CLCON(IKK),COUNT(K),PER
:...FIN
WRITE(IP,450) COUNT(61)
WRITE(IP) TOTBITS,TOT
WRITE(IP,400)
DO (KL=1,310) IBUF(KL)=0
DO (KL=1,200) CALL MTDIO(5,50000K+310,IBUF,IS,IER)
CALL MTDIO(5,60000K,IBUF,IS,IER)
CALL MTDIO(5,60000K,IBUF,IS,IER)
TYPE "TOTAL HITS, TOTAL POINTS ",TOTBITS,TOT

200 FORMAT(I2)
201 FORMAT(I3)
202 FORMAT(2X,"INPUT # OF CORNERS")
203 FORMAT(2X,"INPUT CORNER # ",I3,"J,I")
204 FORMAT(2I4)
205 FORMAT(2X,"LINE #",I5," PROCESSED THERE WERE ",F8.0," HITS")
310 FORMAT(F10.6)
400 FORMAT(////////////////)
410 FORMAT(2X,"NUMBER",2X,"NAME",6X,"APRIORI")
420 FORMAT(4X,I2,2X,4A2,2X,F10.6)
430 FORMAT(2X,"NUMBER NAME RANKING CONSTANT # PIXELS PERCENT")
440 FORMAT(4X,I2,2X,4A2,3X,I2,2X,F10.6,2X,F8.1,3X,F5.1)
450 FORMAT(//,4X,"TOTAL POINTS NOT CLASSIFIED = ",F10.0,//)

STOP
END
```

```

C*****
C
C      CLEANUP
C
C      USED TO CLEAN UP SIGNATURE , VERTEX , OR HIST FILES
C
C      SEQUENCE:  CLEANUP NAMES FILEIN FILEOUT
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION IFIL(20),JFIL(20),KFIL(20),SM(4),COV(4,4),I(6)
C      DIMENSION ICOUNT(4,100),IVX(101),IVY(101),ISW(2)
C      CALL OPEN(1,"COM.CM",1,IERR)
C      CALL COMARG(1,IFIL,ISW,IERR)
C      CALL COMARG(1,IFIL,ISW,IERR)
C      CALL COMARG(1,JFIL,ISW,IERR)
C      CALL COMARG(1,KFIL,ISW,IERR)
C      CALL FOPEN(2,IFIL,"B")
C      CALL FOPEN(4,KFIL,"B")
C      WRITE(10,401)
C      READ(11)ICON
C      CALL FOPEN(3,JFIL,"B")
C      READ BINARY(3)I
C      WRITE BINARY(4)I
C      CALL FCLOSE(3)
C      FOREVER
C      :   CALL FOPEN(3,JFIL,"B")
C      :   READ BINARY(3)I
C      :   ISW=0
C      :   READ(2,100,END=121)NA1,NA2,NA3,NA4
C      :   REPEAT WHILE( ISW.EQ.0)
C      :   :   READ BINARY(3,END=120)NS1,NS2,NS3,NS4
C      :   :   IF(ICON.EQ.1)READ BINARY(3)NP,SM,COV
C      :   :   IF(ICON.EQ.2)
C      :   :   :   READ BINARY(3)K,IYMIN,IYMAX
C      :   :   :   DO (IV=1,K)READ BINARY(3)IV,IVX(IV),IVY(IV)
C      :   :   :   ...FIN
C      :   :   IF(ICON.EQ.3)READ BINARY(3)ICOUNT
C      :   :   IF((NS1.EQ.NA1).AND.(NS2.EQ.NA2))
C      :   :   :   IF((NS3.EQ.NA3).AND.(NS4.EQ.NA4))
C      :   :   :   :   WRITE BINARY(4)NS1,NS2,NS3,NS4
C      :   :   :   :   IF(ICON.EQ.1)WRITE BINARY(4)NP,SM,COV
C      :   :   :   :   IF(ICON.EQ.2)
C      :   :   :   :   :   WRITE BINARY(4)K,IYMIN,IYMAX
C      :   :   :   :   :   DO (IV=1,K)WRITE BINARY(4)IV,IVX(IV),IVY(IV)
C      :   :   :   :   :   ...FIN
C      :   :   :   :   IF(ICON.EQ.3)WRITE BINARY(4)ICOUNT
C      :   :   :   :   ISW=1
C      :   :   :   :   CALL FCLOSE(3)
C      :   :   :   :   ...FIN
C      :   :   :   ...FIN
C      :   ...FIN
C      ...FIN
C      STOP
C      100  FORMAT(4A2)
C      401  FORMAT(2X,"INPUT SWITCH , 1 -SIG , 2 -VER , 3 -HIS")
C      120  STOP - SIGNATURE NOT FOUND
C      121  STOP - NORMAL EXIT
C      END

```

```

*****
C
C      CLUSTA   (SUBROUTINE)
C
C      PURPOSE
C          ADAPTIVE CLUSTERING ALGORITHM, 1-ST PASS THROUGH DEVELOP
C          CLUSTER MEANS USING PERIODIC ELIMINATION TESTS AND
C          MERGER TESTS, ALSO HAS FEATURES OF STRIP FORMULATION AND
C          SEQUENTIAL SEARCH FOR STRIP ASSIGNMENT, 2-ND PASS THROUG
C          DEVELOPS CLASSIFICATION MAP
C
C      DESCRIPTION OF PARAMETERS
C
C      INPUT
C          CALLING SEQUENCE
C              V - VECTORS TO BE CLUSTERED
C              VM - INITIAL CLUSTER MEANS
C              ND - DIMENSION OF VECTORS
C              NV - NUMBER OF VECTORS IN V
C              NVM - NUMBER OF VECTORS IN VM
C              NVMMAX - MAXIMUM NUMBER OF VECTORS ALLOWED IN VM
C              NVG - WEIGHTS FOR CLUSTERS, I-TH VALUE IS NUMBER OF
C                   POINTS IN I-TH CLUSTER
C
C              C,S,RP,R - CLUSTERING DISTANCE MEASURES FOR MERGERS,
C                   STRIP GENERATION, PRIORITY SEARCH, AND
C                   THRESHOLD FOR NEW CLUSTERS RESPECTIVELY
C              NPC,NPT - COUNTER AND THRESHOLD FOR UPDATES TO PRIORITY
C                   LIST
C              PLIST - PRIORITY LIST
C              NEC,NET - COUNTER AND THRESHOLD FOR SMALL CLUSTER
C                   ELIMINATION TESTS
C              NMIN - ELIMINATION THRESHOLD, NUMBER OF POINTS
C              NMC,NMT - COUNTER AND THRESHOLD FOR MERGER TESTS
C              IPASS - PASS NUMBER OR ROUTING FLAG
C              IP - PRINT FLAG, .EQ. 0 NO PRINT, .NE. 0 PRINT MERGERS
C                   AND ELIMINATION MESSAGES
C              JPTP - NUMBER OF POINTS PROCESSED PRIOR TO THIS ENTRY
C              RMINM,RMINV - MEAN AND VARIANCE FOR CLUSTER THRESHOLD
C                   DISTANCES (USED FOR IPASS=2 ONLY), CURRENT
C                   VALUES
C              VMP,VAR - MEANS AND VARIANCES DEVELOPED DURING 2-ND PASS
C                   BASED UPON ACTUAL ASSIGNMENTS - CURRENT VALUES
C
C      OUTPUT
C          CALLING SEQUENCE
C              VM - UPDATED MEANS
C              NVM - NUMBER OF MEANS IN VM
C              NVG - UPDATED WEIGHTS
C              IMG - DEFINES CLASSIFICATION MAP
C              TDIS - THRESHOLD ARRAY
C              NPC,NEC,NMC - UPDATED VALUES
C              RMINM,RMINV - UPDATED VALUES
C              VMP,VAR - UPDATED VALUES
C
C      CREATED AT NASA/JSC   (ASTEP)
C
*****

```

```

*****
SUBROUTINE CLUSTA(V, VM, ND, NV, NVM, NVMMAX, NVG, C, S, RP, R, NPC,
*NPT, PLIST, NEC, NET, NMIN, NMC, NMT, IPASS, IMG, TDIS,
*IP, JPTP, RMINM, RMINV, VMP, VAR)
COMMON/INOUT/INOUT, NIN
INTEGER PLIST
INTEGER TDIS, V
DIMENSION V(ND, NV), VM(ND, NVMMAX), NVG(NVMMAX), PLIST(NVMMAX),
*VS(24), IMG(NV), TDIS(NV)
COMMON/DIST/IDIST
DIMENSION RMINM(NVMMAX), RMINV(NVMMAX)
DIMENSION VMP(ND, NVMMAX), VAR(ND, NVMMAX)
1111 FORMAT(' ND ', I5)
JPT=0
NS=0

```

C
C
C
C

TESTS ON - NEXT POINT, SMALL CLUSTER ELIMINATION, MERGING TIME, AND PLIST UPDATE TIME

```

12 CONTINUE
   IF( IPASS.EQ.2) GO TO 16
   NEC=NEC+NS
   IF( NEC.GE.NET) GO TO 70
14 NMC=NMC+NS
   IF( NMC.GE.NMT) GO TO 80
16 NPC=NPC+NS
   IF( NPC.GE.NPT) GO TO 90
18 JPT=JPT+1
   IF( JPT.GT.(NV-1)) GO TO 100
   JSC=JPT
   JPT=JPT+1

```

C
C
C

STRIP GENERATION

```

19 I=1
20 T=IABS( V(I,JPT) - V(I,JSC) )
   IF ( T.GT.S ) GO TO 22
   I = I + 1
   IF ( I.LE.ND ) GO TO 20
   JPT = JPT + 1
   IF(JPT.LE.NV) GO TO 19
22 JPT = JPT - 1

```

C
C
C

COMPUTE MEAN OF STRIP

```

   NS=JPT-JSC+1
   DO 26 I=1,ND
26  VS(I)=0.
   DO 30 J=JSC,JPT
   DO 28 I=1,ND
28  VS(I)=VS(I)+V(I,J)
30  CONTINUE
   T=FLOAT(NS)
   DO 32 I=1,ND
32  VS(I)=VS(I)/T

```

C
C
C

PRIORITY SEARCH FOR NEAREST CLUSTER

```

34 RMIN=1.E+10
   DO 40 I=1,NVM
   L=PLIST(I)
   D=0.
   IF( IDIST.EQ.2) GO TO 3601
   DO 36 J=1,ND
36  D=D+ABS( VM(J,L)-VS(J) )
   GO TO 3602
3601 CONTINUE
   DO 3603 J=1,ND
3603 D=D+( VM(J,L)-VS(J) )**2
   D=SQRT(D)
3602 CONTINUE
1112 FORMAT( )
   IF( D.GT.RMIN) GO TO 40
   RMIN=D
   J1=L
   IF( RMIN.LT.RP) GO TO 42
40 CONTINUE

```

TEST DISTANCE TO CLUSTER FOR ASSIGNMENT OF STRIP

42 IF(RMIN.GT.R) GO TO 50

FIRST PASS - ASSIGN TO J1

IF(IPASS.EQ.2) GO TO 200
CALL MODIFY(VM(1,J1),VS,NVG(J1),NS,ND)
GO TO 12

FIRST PASS - NEW GROUP, POSSIBLE ELIMINATION OF SMALLEST

50 IF(IPASS.EQ.2) GO TO 210

NVM=NVM+1
J1=NVM
IF(NVM.LE.NVMMAX) GO TO 60
NVM=NVMMAX
NVGP=10000
DO 52 I=2,NVM
IF(NVG(I).GT.NVGP) GO TO 52
IMIN=1
NVGP=NVG(I)

52 CONTINUE

JPTT=JPTP+JPT
900 FORMAT(9H CLUSTER ,13,8H WEIGHT ,14,19H ELIMINATED, JPT = ,14,7H

*VM = 13)
NVG(1)=NVG(1)+NVG(IMIN)
J1=IMIN
60 DO 62 I=1,ND
62 VM(I,J1)=VS(I)
NVG(J1)=NS
NPC=NPT
GO TO 12

SMALL CLUSTER TESTS AND POSSIBLE ELIMINATIONS

70 NEC=0
NPC=NPT
I=1

72 I=I+1

74 IF(I.GT.NVND) GO TO 14
IF(NVG(I).GT.NMIN) GO TO 72

JPTT=JPTP+JPT
IF(IP.NE.0) WRITE(NOUT,900) I,NVG(I),JPTT,NVM
NVG(1)=NVG(1)+NVG(I)
CALL PACK(VM,ND,NVM,I)
CALL PACK(NVG,1,NVM,I)
NVM=NVM-1
GO TO 74

TESTS FOR MERGING AND POSSIBLE MERGING

80 NPC=NPT
NMC=0

MINIMUM DISTANCE BETWEEN CLUSTERS CALCULATION

```

82 I2=NVM-1
   IF(I2.LE.0) GO TO 16
   RMIN=1.E+10
   DO 88 I=1,I2
     I1=I+1
     DO 88 J=I1,NVM
       D=0.
       IF(IDIST.EQ.2) GO TO 8400
       DO 84 L=1,ND
84   D=D+ABS(VM(L,I)-VM(L,J))
       GO TO 8401
8400 CONTINUE
       DO 8402 L=1,ND
8402 D=D+(VM(L,I)-VM(L,J))*2
       D=SQRT(D)
8401 CONTINUE
1007 FORMAT( )
       IF(D.GT.RMIN) GO TO 86
       RMIN=D
       J1=I
       J2=J
86 CONTINUE
88 CONTINUE

```

THRESHOLD TEST

```

IF(RMIN.GT.C) GO TO 16

```

CLUSTER MERGING, J2 INTO J1

```

JPTT=JPTP+JPT
IF(IP.NE.0) WRITE(NOUT,902) J2,NVG(J2),J1,NVG(J1),NVM,JPTT
902 FORMAT(7H MERGER,6H J2 = ,I2,7H NJ2 = ,I4,
*6H J1 = ,I2,7H NJ1 = ,I4,7H NVM = ,I2,7H JPT = ,I4)
CALL MODIFY(VM(1,J1),VM(1,J2),NVG(J1),NVG(J2),ND)
CALL PACK(VM,ND,NVM,J2)
CALL PACK(NVG,1,NVM,J2)
NVM=NVM-1
GO TO 82

```

PLIST UPDATE

```

90 NPC=0
   CALL UPPLT(PLIST,NVG,NVM)
   GO TO 18

```

POSSIBLE SPECIAL CASE FOR LAST POINT

```

100 IF(JPT.NE.NV) RETURN
   DO 102 I=1,ND
102 VS(I)=V(I,NV)
   NS=1
   JSC=JPT
   GO TO 34

```

SECOND PASS - ASSIGN TO J1 AND UPDATE STATISTICS

```

200 CALL THRDST(RMINM(J1),RMINV(J1),NVG(J1),RMIN,NS)
   DO 202 J=JSC,JPT
   CALL SEQST(VMP(1,J1),VAR(1,J1),NVG(J1),ND,V(1,J))
   TDIS(J)=RMIN
202 IMG(J)=J1
   GO TO 12

```

SECOND PASS - ASSGN TO UNASSIGNED

```

210 DO 212 J=JSC,JPT
   TDIS(J)=10000
212 IMG(J)=1
   NVG(1)=NVG(1)+NS
   GO TO 12
END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      COMBINE
C
C      THIS PROGRAM COMBINES SIGNATURES
C      WITH A GIVEN SPECIFIER FROM TWO FILES
C
C      SEQUENCE:  COMBINE FILE1 FILE2 FILE3
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION A(4,20),B(4,4,20),N11(20),N21(20),N31(20),N41(20)
C      DIMENSION IF1(20),IF2(20),IF3(20),C(4,20),D(4,4,20)
C      DIMENSION E(4,30),F(4,4,30),N12(20),N22(20),N32(20),N42(20)
C      DIMENSION N1(30),N2(30),N3(30),N4(30)
C      DIMENSION SM(4),COV(4,4),ISW(2)
C      CALL OPEN(1,"COM.CM",1,IERR)
C      CALL COMARG(1,IF1,ISW,IERR)
C      CALL COMARG(1,IF1,ISW,IERR)
C      CALL COMARG(1,IF2,ISW,IERR)
C      CALL COMARG(1,IF3,ISW,IERR)
C      NSIG=20
C      WRITE(10,103)
C      ND=4
C      IP=10
C      READ(11) ISW
C      CALL FOPEN(2,IF1,"B")
C      CALL FOPEN(3,IF2,"B")
C      CALL FOPEN(4,IF3,"B")
C      READ BINARY(2) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
C      IF( ISW.NE.1) READ BINARY(3) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
C      WRITE(10,101)
C      READ(11,100) NA1,NA2
C      WRITE(10,102)
C      NUN1=2
C      READ(11,104) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
C      NSIG=5
C      CALL CSIG1(A,B,NA1,NA2,NUM1,NSIG,NUN1,N11,N21,N31,N41)
C      WRITE(12,105)
C      NUN2=3
C      NSIG=20
C      IF( ISW.EQ.2)
C      :   CALL CSIG1(C,D,NA1,NA2,NUM2,NSIG,NUN2,N12,N22,N32,N42)
C      :   WRITE(12,106)
C      :...FIN
```

```

DO (K=1,NUM1)
: N1(K)=N11(K)
: N2(K)=N21(K)
: N3(K)=N31(K)
: N4(K)=N41(K)
: DO (K1=1,ND)
: : E(K1,K)=A(K1,K)
: : DO (K2=1,ND)
: : : F(K1,K2,K)=B(K1,K2,K)
: : : ...FIN
: : ...FIN
...FIN
IF(ISW.EQ.1)
: DO (M=1,NUM2)
: : ML=NUM1+M
: : N1(ML)=N12(M)
: : N2(ML)=N22(M)
: : N3(ML)=N32(M)
: : N4(ML)=N42(M)
: : DO (M1=1,ND)
: : : E(M1,ML)=C(M1,M)
: : : DO (M2=1,ND)
: : : : F(M1,M2,ML)=D(M1,M2,M)
: : : : ...FIN
: : : ...FIN
: : ...FIN
...FIN
IF(ISW.EQ.1) ML=NUM1
NP=100
WRITE BINARY(4) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
DO (I=1,ML)
: WRITE BINARY(4) N1(I), N2(I), N3(I), N4(I)
: DO (I1=1,ND)
: : SM(I1)=E(I1,I)
: : DO (I2=1,ND)
: : : COV(I1,I2)=F(I1,I2,I)
: : : ...FIN
: : ...FIN
: WRITE BINARY(4) NP, SM, COV
: WRITE(IP,104) N1(I), N2(I), N3(I), N4(I)
: WRITE(IP) NP, SM, COV
...FIN
100 FORMAT(2A2)
101 FORMAT(2X,"INPUT NA1,NA2")
102 FORMAT(2X,"INPUT TAPE,DATE IN 216 FORMAT")
103 FORMAT(2X,"INPUT # OF FILES")
104 FORMAT(6A2)
105 FORMAT(//2X,"ABOVE ARE SIGNATURES FROM FILE1")
106 FORMAT(//2X,"ABOVE ARE SIGNATURES FROM FILE2")
STOP
END

```

```

C*****
C
C      COPY5
C
C      THIS ROUTINE ADDS 5 SIGNATURE , VERTICES , OR HIST
C      FILES AND PUTS THEM IN A COPYFILE
C
C      SEQUENCE:  COPY5 OUT IN1 IN2 IN3 IN4 IN5
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION IOUT(20), IN1(20), IN2(20), IN3(20), IN4(20), IN5(20)
C      DIMENSION ICOUNT(4,100), I(6), SM(4), COV(4,4), IVX(101)
C      DIMENSION IVY(101), ISW(2)
C      CALL OPEN(1, "COM.CM", 1, IERR)
C      CALL COMARG(1, IOUT, ISW, IERR)
C      CALL COMARG(1, IOUT, ISW, IERR)
C      CALL COMARG(1, IN1, ISW, IERR)
C      CALL COMARG(1, IN2, ISW, IERR)
C      CALL COMARG(1, IN3, ISW, IERR)
C      CALL COMARG(1, IN4, ISW, IERR)
C      CALL COMARG(1, IN5, ISW, IERR)
C      IPR=10
C      WRITE(IPR, 401)
C      READ(11) ICON
C      WRITE(IPR, 402)
C      READ(11) M
C      L=M+2
C      CALL FOPEN(2, IOUT, "B")
C      CALL FOPEN(3, IN1, "B")
C      CALL FOPEN(4, IN2, "B")
C      CALL FOPEN(5, IN3, "B")
C      CALL FOPEN(6, IN4, "B")
C      CALL FOPEN(7, IN5, "B")
C      DO (K=3,L) READ BINARY(K) I
C      WRITE BINARY(2) I
C      TYPE " L = ", L
C      DO (KL=3,L)
C      :   FOREVER
C      :   :   READ BINARY(KL, END=120) NS1, NS2, NS3, NS4
C      :   :   WRITE(IPR, 403) NS1, NS2, NS3, NS4
C      :   :   WRITE BINARY(2) NS1, NS2, NS3, NS4
C      :   :   IF(ICON.EQ.1)
C      :   :   :   READ BINARY(KL) NP, SM, COV
C      :   :   :   TYPE " NP = ", NP
C      :   :   :   WRITE BINARY(2) NP, SM, COV
C      :   :   ...FIN
C      :   :   IF(ICON.EQ.2)
C      :   :   :   READ BINARY(KL) K, IYMIN, IYMAX
C      :   :   :   DO (K1=1,K) READ BINARY(KL) K11, IVX(K1), IVY(K1)
C      :   :   :   WRITE BINARY(2) K, IYMIN, IYMAX
C      :   :   :   DO (K1=1,K) WRITE BINARY(2) K1, IVX(K1), IVY(K1)
C      :   :   ...FIN
C      :   :   IF(ICON.EQ.3)
C      :   :   :   READ BINARY(KL) ICOUNT
C      :   :   :   WRITE BINARY(2) ICOUNT
C      :   :   ...FIN
C      :   ...FIN
120 :   CONTINUE
      ...FIN
401  FORMAT(2X, " INPUT SWITCH - 1 -SIG , 2 - VER , 3 - HIS")
402  FORMAT(2X, " INPUT # OF FILES ")
403  FORMAT(2X, 4A2)
      STOP
      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

C*****
C
C      COUNTY
C
C      DRAWS A POLYGON WHEN GIVEN A STRING OF COORDINATES.
C
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  MICHAEL D. FURMAN
C*****
C
C      INTEGER IC(2,256), IXY(2,2)
C      ACCEPT "# OF POINTS, STRING OF POINTS ", NP, (( IC(I,J), I=1,2), J=1, NP)
C      DO (I=1,2)
C      :   DO (J=1,2) IXY(I,J)=IC(I,1)
C      :   ...FIN
C      DO (J=1,NP)
C      :   DO (K=1,2)
C      :   :   IF ((IC(K,J).LT.IXY(K,1)).AND.(IC(K,J).GE.0)) IXY(K,1)=IC(K,J)
C      :   :   IF (IC(K,J).GT.IXY(K,2)) IXY(K,2)=IC(K,J)
C      :   ...FIN
C      :   ...FIN
C      FAX=( IXY(1,2)-IXY(1,1))/511.0
C      FAY=( IXY(2,2)-IXY(2,1))/511.0
C      WHEN (FAX.GT.FAY) FACT=FAX
C      ELSE FACT=FAY
C      DO (I=1,NP)
C      :   DO (J=1,2) IC(J,I)=(IC(J,I)-IXY(J,1))/FACT
C      :   ...FIN
C      NP1=NP-1
C      DO (I=1,NP1) CALL VECTOR(0, IC(1,I), IC(2,I), IC(1,I+1), IC(2,I+1), 0, 200)
C      CALL VECTOR(0, IC(1,I), IC(2,I), IC(1,1), IC(2,1), 0, 200)
C      STOP
C      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      CRDEM4
C
C      CREATE DEMO FROM DISPLAY IMAGE INCLUDING
C      FUNCTION MEMORY, COLOR MEMORY, AND GRAPHICS
C
C      SEQUENCE: CRDEM4 MTU:F
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMER:  FRED L. THOMPSON
C*****
C      INTEGER ISW(2), INPUT(512), IARRAY(64), IBRAY(0:255), IFLD(10)
C      CALL OPEN (1, "COM.CM", 1, IE)
C      CALL COMARG (1, IFLD, ISW, IE)
C      CALL COMARG (1, IFLD, ISW, IE)
C      CALL MTOPD (3, IFLD, 0, IE)
C      DO (I=1, 512) INPUT(I)=0
C      CALL RCM (0, INPUT(3))
C      ACCEPT "512 OR 256 ", ISZ
C      ACCEPT "TYPE 1 TO RECORD GRAPHICS", IANS
C      TYPE "TYPE A THIRTY CHARACTER DESCRIPTION "
C      TYPE "....."
C      READ (11, 100) (INPUT(I), I=68, 98)
100  FORMAT (30A1)
C      INPUT(1)=ISZ
C      INPUT(2)=IANS
C      CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
C      DO (I=0, 2)
C      :   CALL RFUM (I, INPUT)
C      :   CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
C      :..FIN
C      IEND=ISZ-1
C      ISZ1=ISZ/2
C      DO (IY=0, IEND)
C      :   CALL IMRD (0, IY, INPUT, ISZ1)
C      :   CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
C      :   IF (ISZ.EQ.256)
C      :   :   DO (I=1, 2)
C      :   :   :   CALL IMRD (I, IY, INPUT, ISZ1)
C      :   :   :   CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
C      :   :   :..FIN
C      :   :   IF (IANS.EQ.1)
C      :   :   :   DO (I=0, 3)
C      :   :   :   :   CALL GRD (I, IY, INPUT, 16)
C      :   :   :   :   CALL MTDIO (3, 50000K+ISZ, INPUT, IST, IE, ICNT)
C      :   :   :   :..FIN
C      :   :   :..FIN
C      :..FIN
C      DO (I=1, 2) CALL MTDIO (3, 60000K, INPUT, IST, IE, ICNT)
C      STOP
C      END
```

```

C*****
C
C      CYTAPE
C
C      MAKES FILE INTO A CYBER 74 COMPATABLE FORMAT
C
C      SEQUENCE:  CYTAPE FILE FILE
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  MICHAEL D. FURMAN
C*****
C      DIMENSION S(30), IFILE(10), NS(4), IFILE2(10), ITAP(3), IDAT(3), ISW(2)
C      CALL OPEN(1, "COM.CM", 1, IERR)
C      CALL COMARG(1, IFILE, ISW, IERR)
C      CALL COMARG(1, IFILE, ISW, IERR)
C      CALL COMARG(1, IFILE2, ISW, IERR)
C      CALL FOPEN(2, IFILE, "B")
C      CALL FOPEN(3, IFILE2, "B")
C      READ BINARY(2, END=205) ITAP, IDAT
C      FOREVER
C      : DO (I=1,30) S(I)=0.
C      : READ BINARY(2, END=205) NS
C      : READ BINARY(2, END=205) NP, (S(I), I=1, 20)
C      : WRITE BINARY(3) NS
C      : WRITE(3)
C      : WRITE(3, 100) NP, (S(I), I=1, 20)
C      :...FIN
C      STOP
205 STOP FINISHED
100 FORMAT(1X, I4, 5(4F12.8, /))
END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      DISDEM4
C
C      DISPLAYS DEMO TAPES TO COMTAL COLOR DISPLAY
C      WITH GRAPHICS COLOR MEMORY AND FUNCTION MEMORY
C
C      SEQUENCE:  DISDEM4 MTU:F
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  FRED L. THOMPSON
C*****
C      INTEGER ISW(2), INPUT(512), IFLD(10)
C      CALL OPEN (1, "COM.CM", 1, IE)
C      CALL COMARG (1, IFLD, ISW, IE)
C      CALL COMARG (1, IFLD, ISW, IE)
C      CALL MTOPD (3, IFLD, 0, IE)
C      FOREVER
C      :   CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
C      :   ISZ=INPUT(1)
C      :   IANS=INPUT(2)
100 :   WRITE (10, 100) (INPUT(I), I=68, 98)
C      :   FORMAT (1X, 30A1)
C      :   IF (ISZ.EQ.512) PAUSE SET BIC FOR 512 HIT RETURN
C      :   IF (ISZ.EQ.256) PAUSE SET BIC FOR 256 HIT RETURN
C      :   CALL WCM (0, INPUT(3))
C      :   DO (I=0, 2)
C      :   :   CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
C      :   :   CALL WFUM (1, INPUT)
C      :   :   ...FIN
C      :   IEND=ISZ-1
C      :   ISZ1=ISZ/2
C      :   DO (IY=0, IEND)
C      :   :   CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
C      :   :   CALL IMWR (0, IY, INPUT, ISZ1)
C      :   :   IF (ISZ.EQ.256)
C      :   :   :   DO (I=1, 2)
C      :   :   :   :   CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
C      :   :   :   :   CALL IMWR (1, IY, INPUT, ISZ1)
C      :   :   :   :   ...FIN
C      :   :   :   IF (IANS.EQ.1)
C      :   :   :   :   DO (I=0, 3)
C      :   :   :   :   :   CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
C      :   :   :   :   :   CALL GWR (1, IY, INPUT, 16)
C      :   :   :   :   :   ...FIN
C      :   :   :   :   ...FIN
C      :   :   :   ...FIN
C      :   :   ...FIN
C      :   ...FIN
C      :   CALL MTDIO (3, 0, INPUT, IST, IE, ICNT)
C      :   PAUSE HIT RETURN FOR NEXT FILE CTRL A TO END
C      ...FIN
C      STOP
C      END
```


USED WITH ELIP TO DRAW 2-D ELLIPSES OF SIGNATURES
ON THE VERSATEC

PROGRAMMER: G. DAVID GENTRY

68

```

C*****
C
C      DTAPE
C
C      GENERAL TAPE DUMPING ROUTINE FOR READING AND
C      DEBUGGING FOREIGN TAPES
C
C      SEQUENCE:  DTAPE MTU:F
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  MICHAEL D. FURMAN
C*****
C      INTEGER IDATA(4110), IOUT(8220), IFLD(10), ISW(2)
C      CALL OPEN(1, "COM.CM", 1, IERR)
C      CALL COMARG(1, IFLD, ISW, IERR)
C      CALL COMARG(1, IFLD, ISW, IERR)
100  FORMAT(1X, 20I6)
101  FORMAT(1X, 25I4)
102  FORMAT(1X, 100A1)
103  FORMAT(1X, 10(0I8, 1HK))
104  FORMAT(1X, 20(0I5, 1HK))
C      ACCEPT "WORDS? (1), OR BYTES? (0) OUTPUT ", IP
C      ACCEPT "BASE: TEN= 1, EIGHT= 0 ", IN
C      ACCEPT "SKIP HOW MANY RECORDS? ", ISKP
C      ACCEPT "DUMP HOW MANY BLOCKS OF DATA ", IBLOK
C      IF ((IP.EQ.0).AND.(IN.EQ.1))
C      :   TYPE "TYPE: NUMERIC= 0 ; ASCII= 1 ; EBCDIC= 2 "
C      :   READ(11) IQ
C      :   ...FIN
C      CALL MTOPD(3, IFLD, 0, IERR)
C      IF (ISKP.GT.0)
C      :   DO (I=1, ISKP) CALL MTDIO(3, 0, IDATA, IS, IE, IC)
C      :   ...FIN
C      DO (K1=1, IBLOK)
C      :   DO (I=1, 4110) IDATA(I)=0
C      :   DO (I=1, 8220) IOUT(I)=0
C      :   CALL MTDIO(3, 0, IDATA, IS, IE, IC)
C      :   TYPE "RECORDS PER BLOCK= ", IC
C      :   IF (IP.EQ.0)
C      :   :   CALL UPACS(IDATA, IOUT, IC)
C      :   :   IF (IQ.EQ.2) CALL EBCDIC(IOUT, IOUT, ID)
C      :   :   IC=IC*2
C      :   ...FIN
C      :   WHEN (IQ.NE.0) WRITE(12, 102) (IOUT(I), I=1, IC)
C      :   ELSE
C      :   :   IF ((IN.EQ.1).AND.(IP.EQ.1)) WRITE(12, 100) (IDATA(I), I=1, IC)
C      :   :   IF ((IP.EQ.1).AND.(IN.EQ.0)) WRITE(12, 103) (IDATA(I), I=1, IC)
C      :   :   IF ((IP.EQ.0).AND.(IN.EQ.1)) WRITE(12, 101) (IOUT(I), I=1, IC)
C      :   :   IF ((IP.EQ.0).AND.(IN.EQ.0)) WRITE(12, 104) (IOUT(I), I=1, IC)
C      :   ...FIN
C      :   WRITE(12)
C      :   ...FIN
C      STOP
C      END

```

```

*****
EBCDIC  (SUBROUTINE)

```

```

CONVERTS EBCDIC CODE INTO ASCII

```

```

SEQUENCE:  EBCDIC(INPUT ARRAY, OUTPUT ARRAY, ARRAY DIMENSIONS)

```

```

*****
CREATED AT GEORGIA TECH EES

```

```

PROGRAMMER:  MICHAEL D. FURMAN
*****

```

```

SUBROUTINE EBCDIC(INPUT, IOUT, IPR)
INTEGER INPUT(IPR), IOUT(IPR)
COMMON/DUM1/ICNTAB(263)
DATA ICNTAB/1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
11H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
11H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
11H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
21H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
21H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
21H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
31H?, 1H-, 1H/, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
31H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
31H?, 1H?, 1H', 1H=, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
41H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
41H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
41H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
51H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?, 1H?,
51HA, 1HB, 1HC, 1HD, 1HE, 1HF, 1HG, 1HH, 1HI, 1H?, 1H?, 1H?, 1H?,
51H?, 1H?, 1HJ, 1HK, 1HL, 1HM, 1HN, 1HO, 1HP, 1HQ, 1HR, 1H?, 1H?, 1H?,
61H?, 1H?, 1H?, 1H?, 1H?, 1HS, 1HT, 1HU, 1HV, 1HW, 1HX, 1HY, 1HZ, 1H?,
61H?, 1H?, 1H?, 1H?, 1H?, 1H0, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8,
61H9, 1H?, 1H?, 1H?, 1H?, 1H?/

```

```

DO (I=1, IPR)
:   K1=INPUT(I)+1
:   WHEN (K1.LE.256)
:     :   IOUT(I)=ICNTAB(K1)
:     :   ...FIN
:   ELSE IOUT(I)=ICNTAB(I)
:   ...FIN
END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      EDIST  (SUBROUTINE)
C
C      PURPOSE:
C      COMPUTES THE DISTANCE BETWEEN TWO VECTORS
C
C      DESCRIPTION OF PARAMETERS
C
C      CALLING SEQUENCE
C      INPUT
C      V1,V2 - TWO VECTORS
C      ND - DIMENSION OF THE VECTORS
C
C      OUTPUT
C      CALLING SEQUENCE
C      D - CONTAINS DISTANCE BETWEEN V1 AND V2
C*****
C      CREATED AT NASA/JSC  (ASTEP)
C*****
C      SUBROUTINE EDIST ( V1, V2, ND, D )
C      DIMENSION V1(ND),V2(ND)
C      D = 0.
C      DO 10 I=1,ND
10  D = D + (V1(I)-V2(I))**2
C      D = SQRT(D)
C      RETURN
C      END
```

```

C*****
C
C      EIGEN   (SUBROUTINE)
C
C      COMPUTES EIGENVALUES AND EIGENVECTORS OF REAL
C      SYMMETRICAL MATRIX
C
C*****
C      CREATED AT NASA/JSC   (ASTEP)
C*****
SUBROUTINE EIGEN(AA,N,NV,A,E,R)
  DIMENSION AA(N,N),R(N,N),E(N),A(N,N)
202  FORMAT(/" SUBROUTINE EIGEN ERROR RETURN-FINAL NORM HAS NOT
1BEEN REACHED AFTER 100 ITERATIONS"/)
  DO (I=1,N)
    : DO (J=1,N)
    :   : A(I,J)=AA(I,J)
    :   : ...FIN
  : ...FIN
  RANGE=1.E-6
  IF(MV-1) 10,25,10
10  DO (I=1,N)
    : DO (J=1,N)
    :   : IF(I.EQ. J) GO TO 15
    :   : R(I,J)=0.
    :   : GO TO 20
    :   : R(I,J)=1.
15  :   : CONTINUE
20  :   : ...FIN
    : ...FIN
25  ANORM=0.
    DO (I=1,N)
    : DO (J=1,N)
    :   : IF(I.EQ. J) GO TO 35
    :   : ANORM=ANORM+A(I,J)*A(I,J)
35  :   : CONTINUE
    :   : ...FIN
    : ...FIN
40  IF(ANORM) 165,165,40
    ANORM=SQRT(ANORM)
    ANRMX=ANORM*RANGE
    ICNT=0
    IND=0
    THR=ANORM
45  THR=THR/FLOAT(N)
    ICNT=ICNT+1
    IF(ICNT.EQ. 100) GO TO 200
50  MQ=2
55  MP=1
    X=.5*(A(MP,MP)-A(MQ,MQ))
62  IF(ABS(A(MP,MQ))-THR) 138,65,65
65  IND=1
    X=.5*(A(MP,MP)-A(MQ,MQ))
    Y=-A(MP,MQ)/SQRT(A(MP,MQ)*A(MP,MQ)+X*X)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

70      IF(X) 70,75,75
75      Y=-Y
      SINX=Y/SQRT(2.*(1.+(SQRT(1.-Y*Y))))
      SINX2=SINX*SINX
      COSX=SQRT(1.-SINX2)
      COSX2=COSX*COSX
      SINCS=SINX*COSX
      XX=A(MP,MP)
      YY=A(MQ,MQ)
      ZZ=A(MP,MQ)
      DO (I=1,N)
        : X=A(I,MP)*COSX-A(I,MQ)*SINX
        : A(I,MQ)=A(I,MP)*SINX+A(I,MQ)*COSX
        : A(I,MP)=X
        : IF(MV-1) 120,125,120
120      : X=R(I,MP)*COSX-R(I,MQ)*SINX
        : R(I,MQ)=R(I,MP)*SINX+R(I,MQ)*COSX
        : R(I,MP)=X
125      : CONTINUE
      ...FIN
      X=2.*ZZ*SINCS
      Y=(XX*COSX2)+(YY*SINX2)-X
      X=(XX*SINX2)+(YY*COSX2)+X
      A(MP,MP)=Y
      A(MQ,MQ)=X
      A(MP,MQ)=(XX-YY)*SINCS+ZZ*(COSX2-SINX2)
      A(MQ,MP)=0.
      DO (I=1,N)
        : A(MP,I)=A(I,MP)
        : A(MQ,I)=A(I,MQ)
      ...FIN
138     IF(MP.NE.(MQ-1))GO TO 140
        IF(MQ.NE.N) GO TO 145
        IF(IND.NE.1)GO TO 160
        IND=0
        GO TO 50
140     MP=MP+1
        GO TO 62
145     MQ=MQ+1
        GO TO 55
160     IF(THR - ANRMX) 165,165,45
165     CONTINUE
        DO (I=1,N)
          : DO (J=1,N)
          : : IF(I.NE.J) GO TO 190
          : : E(I)=A(I,J)
190      : : CONTINUE
          : ...FIN
        ...FIN
        GO TO 210
200     WRITE(10,202)
210     RETURN
      END

```

ELIP

DRAWS 2-D ELLIPSES OF COVARIANCE MATRICIES

SEQUENCE: ELIP REQFIL SIGFILC

CREATED AT GEORGIA TECH EES

PROGRAMMER: G. DAVID GENTRY

```
DIMENSION REQ(30),SIGFIL(30),E(2,4),R(2,2),D(2,2),IORDER(20)
DIMENSION COVIN(4,4,20),SMIN(4,20),N1(20),N2(20),N3(20),N4(20)
DIMENSION COV1(2,2),SM1(2),ISW(2)
CALL OPEN(1,"COM.CM",1,IERR)
CALL COMARG(1,REQ,ISW,IERR)
CALL COMARG(1,REQ,ISW,IERR)
CALL COMARG(1,SIGFIL,ISW,IERR)
CALL FOPEN(2,REQ,"B")
IO=10
ACCEPT "INPUT NUMBER OF ELLIPSES PER PLOT ",NUMS
IIN=11
ND=2
NSIG=20
CALL MODE(1,1.,.5,-1.)
MSIG=NSIG/NUMS
DO(JV=1,NSIG)
: NSIG=NUMS
: KO=0
: CALL CSIG(SMIN,COVIN,NSIG,2,3,SIGFIL,IORDER,N1,N2,N3,N4)
: DO (JJ=1,3)
: : K=JJ+1
: : DO (KK=K,4)
: : : AX1=JJ
: : : AX2=KK
: : : ID=0
: : : DO (II=1,NSIG)
: : : : SM1(1)=255.-SMIN(JJ,ID)
: : : : SM1(2)=255.-SMIN(KK,ID)
: : : : COV1(1,1)=COVIN(JJ,JJ,II)
: : : : COV1(1,2)=COVIN(JJ,KK,II)
: : : : COV1(2,1)=COVIN(KK,JJ,II)
: : : : COV1(2,2)=COVIN(KK,KK,II)
: : : : CALL FACANL(COV1,SM1,ND,E,R,D)
: : : : CALL DRAWER(SM1,E,COV1,AX1,AX2,R,ID)
: : : : ID=ID+1
: : : ...FIN
: : : KO=KO+1
: : : IF(KO.EQ.1)
: : : : CALL DRAW(0.,0.,1,0000)
: : : : CALL DRAW(-.5,8.5,1,1001)
: : : : ...FIN
: : : IF(KO.EQ.2)
: : : : KO=0
: : : : CALL DRAW(0.,0.,1,0000)
: : : : CALL DRAW(0.,-9.5,1,1001)
: : : : CALL DRAW(0.,0.,1,9000)
: : : : ...FIN
: : ...FIN
: ...FIN
CALL DRAW(0,0,0,9999)
STOP
END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```

C*****
C
C      FACANL  (SUBROUTINE)
C
C      USED WITH ELIP TO DO FACTOR ANALYSIS OF COV. MATRIX
C*****
C
C      CREATED BY NASA JSC
C*****
SUBROUTINE FACANL(COV,SM,ND,E,R,D)
DIMENSION COV(ND,ND),SM(ND),E(ND,4),R(ND,ND),D(ND,ND)
MV=0
CALL EIGEN(COV,ND,MV,D,E,R)
ND1=ND-1
IF(ND1.EQ. 0)RETURN
DO (J=1,ND1)
:   EMAX=0.
:   DO (I=J,ND)
:   :   IF(E(I,1).LT. EMAX) GO TO 50
:   :   EMAX=E(I,1)
:   :   IS=I
:   :   CONTINUE
:   ...FIN
:   DUM=E(J,1)
:   E(J,1)=E(IS,1)
:   E(IS,1)=DUM
:   DO (I=1,ND)
:   :   DUM=R(I,J)
:   :   R(I,J)=R(I,IS)
:   :   R(I,IS)=DUM
:   ...FIN
:   ...FIN
DUM=0.
DO (I=1,ND)
:   DUM=DUM+E(I,1)
:   ...FIN
DUM=1./DUM
EMAX=0.
DO (I=1,ND)
:   E(I,2)=DUM*E(I,1)
:   E(I,3)=EMAX+E(I,2)
:   EMAX=E(I,3)
:   ...FIN
DUM=0.
DO (I=1,ND)
:   DUM=DUM+SM(I)**2
:   ...FIN
DUM=SQRT(DUM)
DO (J=1,ND)
:   E(J,4)=0.
:   DO (I=1,ND)
:   :   E(J,4)=E(J,4)+SM(I)*R(I,J)
:   ...FIN
:   E(J,4)=E(J,4)/DUM
:   E(J,4)=57.29578*ACOS(E(J,4))
:   ...FIN
RETURN
END

```

50


```

C*****
C
C      FILTER1
C
C      PERFORMS A HIGH-PASS FILTER ON DISPLAY
C
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  MICHAEL D. FURMAN
C*****
C      INTEGER IN(512,3), IMAG1(256), IMAG2(512), IMAG3(256)
C      INTEGER IN2(512,3), IN3(512,3)
C      ACCEPT "512 OR 256 MODE ", IMODE
C      IMOM1= IMODE-1
C      INOM2= IMODE-2
C      IANS=0
C      IF (IMODE.EQ.256)
C      :   ACCEPT "ALL 3 IMAGES AT ONCE? (1=YES) ", IANS
C      :   IF (IANS.NE.1) ACCEPT "IMAGE NUMBER(0-2) ", IM2
C      :   ...FIN
C      DO (IM=0,2)
C      :   IF ((IANS.NE.1).AND.(IMODE.NE.512)) IM= IM2
C      :   M3=2
C      :   M4=3
C      :   DO (I=0,1)
C      :   :   CALL IMREAD(IM,I,IMAG1,256)
C      :   :   CALL UPAC8(IMAG1,IN(1,(I+1)),256)
C      :   :   ...FIN
C      :   DO (M5=2,IMOM2)
C      :   :   M5M1=M5-1
C      :   :   CALL IMREAD(IM,M5,IMAG1,256)
C      :   :   CALL UPAC3(IMAG1,IN(1,M4),256)
C      :   :   DO (I=1,MODE) IMAG2(I)=0
C      :   :   DO (J1=1,3)
C      :   :   :   DO (K=2,IMOM1) IMAG2(K)=IMAG2(K)+IN(K-1,J1)+IN(K,J1)+IN(K+1
C      :   :   :   ...FIN
C      :   :   DO (I=1,IMODE)
C      :   :   :   IMAG2(I)=(IN(1,M3)*2)-((IMAG2(I)/9.0)+0.5)
C      :   :   :   IF (IMAG2(I).LT.0) IMAG2(I)=0
C      :   :   :   IF (IMAG2(I).GT.255) IMAG2(I)=255
C      :   :   :   ...FIN
C      :   :   CALL PAC8(IMAG2,IMAG3,512)
C      :   :   CALL IMWRITE(IM,M5M1,IMAG3,256)
C      :   :   WHEN (M4.EQ.3) M4=1
C      :   :   ELSE M4=M4+1
C      :   :   WHEN (M3.EQ.3) M3=1
C      :   :   ELSE M3=M3+1
C      :   :   ...FIN
C      :   IF ((IANS.NE.1).OR.(IMODE.EQ.512)) STOP
C      :   ...FIN
C      CALL BACK
C      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

C*****
C
C      GETPOLY
C
C      THIS PROGRAM SUMS AREAS OF ALL MAXIMUM LIKLIHOOD CLASS
C      WITHIN A SPECIFIED POLYGON
C      POLYGON DATA IS THEN DUMPED TO TAPE
C
C      SEQUENCE: GETPOLY INPUT MTU:F MTU:F
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION COUNT(60),JBUF(3300),IVX(201),IVY(201)
C      DIMENSION LA(233),KBUF(3300)
C      DIMENSION IFIL(20),ITAP(20),JTAP(20),ISW(2)
C      IIN=11
C      IIO=10
C      CALL OPEN(1,"COM.CM",1,IERR)
C      CALL COMARG(1,IFIL,ISW,IERR)
C      CALL COMARG(1,IFIL,ISW,IERR)
C      CALL COMARG(1,ITAP,ISW,IERR)
C      CALL COMARG(1,JTAP,ISW,IERR)
C      CALL FOPEN(IIN,IFIL,"B")
C      CALL MTOPD(3,ITAP,0,IE)
C      CALL MTOPD(4,JTAP,0,IE)
C      WRITE(IIO,400)
C      DO (I=1,3300)KBUF(I)=0
C      DO (I=1,133)LA(I)=0
400  FORMAT(2X,"INPUT 1 FOR OUTPUT OF POLYGON TO TAPE")
C      DO (I=1,60) COUNT(I)=0.0
C      READ(IIN) IOUT
C      WRITE(IIO) IOUT
C      ICUT=10000
C      WRITE(IIO,401)
401  FORMAT(2X,"INPUT 1 TO USE A THRESHOLD")
C      READ(IIN) IZ
C      WRITE(IIO) IZ
C      IF (IZ.EQ.1)
C      :   WRITE(IIO,402)
402  :   FORMAT(2X,"INPUT THRESHOLD ")
C      :   READ(IIN) ICUT
C      :   WRITE(IIO) ICUT
C      :..FIN
C      ACCEPT "  START LINE OF CLASSIFIED DATA ? ",ISTART
C      IYMIN=10000
C      IXMIN=10000
C      IYMAX=0
C      READ(IIN) NV
C      WRITE(IIO) NV
201  FORMAT(2I2)
C      DO (I=1,NV)
C      :   READ(IIN,206) IVX(I),IVY(I)
C      :   IVY(I)=IVY(I)-ISTART+1
C      :   IF (IVY(I).LT.1) IVY(I)=1
C      :   IF (IVX(I).LT.1) IVX(I)=1
C      :   WRITE(IIO) IVX(I),IVY(I)
C      :   IF (IVY(I).LT.IYMIN) IYMIN=IVY(I)
C      :   IF (IVX(I).LT.IXMIN) IXMIN=IVX(I)
C      :   IF (IVY(I).GT.IYMAX) IYMAX=IVY(I)
C      :..FIN

```

```

IVX(NV+1)=IVX(1)
IVY(NV+1)=IVY(1)
IDEL=IYMAX-IYMIN+1
WRITE(110) IDEL
IF(IOUT.EQ.1)
:   JBUF(1)=IDEL
:   CALL MTDIO(4,50000K+3300,JBUF,IS,IE)
...FIN
ISKIP=(IYMIN-1)*2+1,
IF(ISKIP.GT.0)
206 :   FORMAT(214)
:   DO (K=1,ISKIP)
:   :   CALL MTDIO(3,0,JBUF,IS,IE)
:   ...FIN
...FIN
ISK=IYMIN-1
DO (J=1, IDEL)
:   DO (I=1,3300) JBUF(I)=60
:   DO (I=1,3300) KBUF(I)=10000
:   CALL MTDIO(3,0,JBUF,IS,IE)
:   CALL MTDIO(3,0,KBUF,IS,IE)
:   ISW=2
:   CALL POLY2(IYMIN+J-1,ISW,IVX,IVY,NV,LA)
:   NSTR=LA(1)
:   JV=1
:   WRITE(10) NSTR
:   DO (NSEQ=1,NSTR)
:   :   LSUB=2*NSEQ
:   :   L2=LSUB+1
:   :   JS=LA(LSUB)
:   :   IF(JS.LT.1) JS=1
:   :   JF=LA(L2)
:   :   WRITE(110,28) JS,JF,JV
28 :   :   FORMAT(2X,"JS,JF,JV",3I10)
:   :   JS1=JS-1
:   :   IF(JS1.NE.0)
:   :   :   DO (K2=JV,JS1) JBUF(K2)=60
:   :   :   DO (K3=JV,JS1) KBUF(K3)=10000
:   :   ...FIN
:   :   JV=JF+1
:   :   DO (J1=JS,JF)
:   :   :   IF(KBUF(J1).GT.ICUT) JBUF(J1)=60
:   :   :   LS1=JBUF(J1)
:   :   :   IF(LS1.GT.60) LS1=60
:   :   :   COUNT(LS1)=COUNT(LS1)+1.
:   :   ...FIN
:   ...FIN
:   JF1=JF+1
:   DO (J1=JF1,3300) JBUF(J1)=60
:   DO (J1=JF1,3300) KBUF(J1)=10000
:   IF(IOUT.EQ.1)
:   :   CALL MTDIO(4,50000K+3300,JBUF,IS,IE)
:   :   CALL MTDIO(4,50000K+3300,KBUF,IS,IE)
:   :   WRITE(10,211) J
211 :   :   FORMAT(2X,"LINE # ",I5," PROCESSED")
:   ...FIN
...FIN
DO (K=1,60)
:   ACRE=COUNT(K)*1.05
:   WRITE(110,100) K,COUNT(K),ACRE
100 :   FORMAT(2X,"CLASS # ",I10," SIZE ",F11.0," ACRES ",F11.0)
...FIN
CALL MTDIO(4,60000K,JBUF,IS,IE)
CALL MTDIO(4,60000K,JBUF,IS,IE)
STOP
END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

C*****
C
C      GSIG  (SUBROUTINE)
C
C      GETS CLASSIFICATION SIGNATURES
C      NUN1 IS INPUT FILE (NAMES OF SIGNATURES TO BE RETRIEVED)
C      NUN2 IS SIGFILE FROM WHICH SIGNATURES WILL BE TAKEN
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
SUBROUTINE GSIG(A,B,NSIG,ITEMSIG,N1,N2,N3,N4,APR)
DIMENSION A(4,60),B(4,4,60),SM(4),COV(4,4),ITEMSIG(34)
DIMENSION N1(60),N2(60),N3(60),N4(60),APR(60)
IP=12
NUN1=2
NUN2=4
DO (I=1,NSIG)
:  READ(NUN1,100)NA1,NA2,NA3,NA4,APRI
:  N1(I)=NA1
:  N2(I)=NA2
:  N3(I)=NA3
:  N4(I)=NA4
:  APR(I)=APRI
:  CALL FOPEN(NUN2,ITEMSIG,"B")
:  READ BINARY(NUN2)ITAP1,ITAP2,ITAP3,IDAT1,IDAT2,IDAT3
:  ISET=0
:  WHILE(ISET.EQ.0)
:  :  READ BINARY(NUN2,END=120)NS1,NS2,NS3,NS4
:  :  READ BINARY(NUN2)NP,SM,COV
:  :  IF(NS1.EQ.NA1.AND.NS2.EQ.NA2)
:  :  :  IF(NS3.EQ.NA3.AND.NS4.EQ.NA4)
:  :  :  :  DO (K=1,4)
:  :  :  :  :  A(K,I)=SM(K)
:  :  :  :  :  DO (L=1,4)
:  :  :  :  :  :  B(K,L,I)=COV(K,L)
:  :  :  :  :  ...FIN
:  :  :  :  ...FIN
:  :  :  :  ISET=1
:  :  :  :  CALL FCLOSE(NUN2)
:  :  :  :  ...FIN
:  :  ...FIN
:  ...FIN
RETURN
100  FORMAT(4A2,F10.8)
120  STOP  FILE NOT FOUND
END

```

```

C*****
C      HISTO      (SUBROUTINE)
C      THIS SUBROUTINE CALCULATES AND DISPLAYS A HISTOGRAM FOR A
C      SPECIFIED DATA SET - 4 CHANNELS = 4 HISTOGRAMS
C*****
C      CREATED AT GEORGIA TECH EES
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
SUBROUTINE HIST(NUM)
  DIMENSION ICOUNT(4,100),MAX(4),MIN(4)
  COMMON/DTRANS/IMODL(256),IX(512,4),IDUM(512)
  COMMON /HISTO/ ICOUNT
  ND=4
  DO (IJ=1,100)
    : DO (JI=1,ND)
    : : ICOUNT(JI,IJ)=0
    : :...FIN
  :...FIN
  DO (I=1,ND)
    : DO (K=1,NUM)
    : : J=IX(K,I)
    : : IF(J.EQ.0)J=99
    : : IF(J.GT.99)J=99
    : : ICOUNT(I,J)=ICOUNT(I,J)+1
    : :...FIN
  :...FIN
  DO (KCH=1,ND)
    : MIN(KCH)=1000
    : MAX(KCH)=0
    : DO (KK=1,60)
    : : IF(ICOUNT(KCH,KK).LT.MIN(KCH))MIN(KCH)=ICOUNT(KCH,KK)
    : : IF(ICOUNT(KCH,KK).GT.MAX(KCH))MAX(KCH)=ICOUNT(KCH,KK)
    : :...FIN
    : IF(MAX(KCH).EQ.0)MAX(KCH)=1
    : DO (JJ=1,60)
    : : ICOUNT(KCH,JJ)=(ICOUNT(KCH,JJ)*20)/MAX(KCH)
    : :...FIN
  :...FIN
  DO (L=1,20)
    : LI=256-L
    : DO (K=1,512) IDUM(K)=0
    : DO (KCH=1,4)
    : : ISUB=(KCH-1)*60+NJ
    : : IF(ICOUNT(KCH,NJ).GT.L)IDUM(ISUB)=140
    : :...FIN
  :...FIN
  IDUM(1)=140
  IDUM(60)=140
  IDUM(120)=140
  IDUM(180)=140
  CALL CPACK1(IDUM,IMODL,0)
  WHEN(L.NE.1)CALL INWRITE(0,LI,IMODL,256)
  ELSE
    : DO (L2=1,255)
    : : IMODL(L2)=106214K
    : :...FIN
    : CALL INWRITE(0,LI,IMODL,256)
  :...FIN
  ACCEPT "HISTO ENDED, ENTER 1 TO CONTINUE : ",ISTOP
  RETURN
END

```

81

```
100  FORMAT(2X,"TAPE IDENTIFIER - ",3I2)
101  FORMAT(2X,"SIGNATURE TAKEN ON - ",I2,"/",I2,"/",I2)
102  FORMAT(1X,4A2)
104  FORMAT(123A1)
106  FORMAT(4A2)
105  FORMAT(///// )
120  STOP
      END
```

INITCL (SUBROUTINE)

PURPOSE:

INITIALIZES THE MEAN VECTORS AND WEIGHTS FOR THE CLUSTERING ALGORITHMS

DESCRIPTION OF PARAMETERS

INPUT

\$NAMELIST-SININIT

VM - VECTOR MEANS

NVM - NUMBER OF VECTORS IN VM

NVG - WEIGHT ASSOCIATED WITH EACH MEAN

KNO - DIMENSION OF EACH VECTOR IN VM

HOLLERITH OPTIONS

ZERO - SETS INITIAL MEANS AND WEIGHTS EQUAL TO ZERO

OLD - RETURNS WITH PREVIOUSLY COMPUTED MEANS AND WEIGHTS

NEW - ALLOW MEANS AND WEIGHTS TO BE INPUT

OUTPUT

CALLING SEQUENCE

VM - VECTOR MEANS

NVM - NUMBER OF VECTORS IN VM

NVG - WEIGHT ASSOCIATED WITH EACH MEAN

PRINT

DISPLAYS INPUT IF CARD OPTION IS NEW

REMARKS AND RESTRICTIONS

AFTER NAMELIST INPUT A CARD WITH THE WORD YES IS REQUIRED

INDICATING CORRECT INPUTS IF NOT REPEAT NAMELIST INPUTS

CREATED AT NASA/JSC (ASTEP)

```

SUBROUTINE INITCL(VM,NVG,NVM,KNO)
  DIMENSION VM(240),NVG(20)
  COMMON/INOUT/IOUT,INP
1040 WRITE(IOUT,1050)
1050 FORMAT(38H CHOOSE VALUES FOR INITIALIZATION FROM
  WRITE(IOUT,1055)
1055 FORMAT(15H ZERO OLD NEW)
  READ(INP,1030) IAN
1030 FORMAT(A6)
  IF(IAN.EQ.4HZERO) GO TO 1100
  IF(IAN.EQ.3HOLD) GO TO 1200
  IF(IAN.EQ.3HNEW) GO TO 1150
  WRITE(IOUT,1060) IAN
1060 FORMAT(1H ,A6,23H IS NOT A VALID CHOICE.)
  GO TO 1040
1100 CONTINUE
  DO 51 I=1,240
51 VM(I)=0.0
  DO 52 I=1,20
52 NVG(I)=0
  NVM=1
  GO TO 1200
1150 CONTINUE
1170 WRITE(IOUT,1160)
1160 FORMAT(21H SININIT VM,NVG,NVM)
  WRITE(IOUT,1022) NVM,(NVG(I),I=1,NVM)
1022 FORMAT(7H NVM = ,I2,7H NVG = ,20I6)
  CALL MATPRT(VM,KNO,KNO,NVM,5HMEANS)
  WRITE(IOUT,1020)
1020 FORMAT(32H TYPE YES IF INPUTS ARE CORRECT.)
  READ(INP,1030) IAN
  IF(IAN.NE.3HYES) GO TO 1170
1200 CONTINUE
  RETURN
  END

```



```

C*****
C
C      INPUTSIG
C
C      ALLOWS MANUAL INPUT OF SIGNATURE FILE
C
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION COV(4,4),XM(4)
C      CALL OPEN(3,"INSG",0,IE)
C      CALL FOPEN(4,"HSIG","B")
C      READ(3)NSIG
C      ID1=01
C      ID2=22
C      ID3=77
C      IT1=99
C      IT2=99
C      IT3=99
C      WRITE BINARY(4) IT1,IT2,IT3, ID1, ID2, ID3
C      DO (L=1,NSIG)
C      :   WRITE(10,101)
C      :   READ(3,100)NA1,NA2,NA3,NA4
C      :   WRITE(10,102)
C      :   READ(3)NP
C      :   WRITE(10,103)
C      :   READ(3)XM
C      :   DO (K=1,4)
C      :   :   WRITE(10,104)K
C      :   :   READ(3)(COV(K,J),J=1,4)
C      :   :   ...FIN
C      :   WRITE BINARY(4)NA1,NA2,NA3,NA4
C      :   WRITE BINARY(4)NP,XM,COV
C      :   ...FIN
100  FORMAT(4A2)
101  FORMAT(2X,"INPUT NA1,NA2,NA3,NA4")
102  FORMAT(2X,"INPUT NUMBER OF POINTS")
103  FORMAT(2X,"INPUT (XM(I),I=1,4)")
104  FORMAT(2X,"INPUT ROW # ",I1," OF COV - (COV(K,J),J=1,4)")
C      STOP
C      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      LINBOX  (SUBROUTINE)
C
C      USED TO CALCULATE EXTREMES FOR A LINEAR CLASSIFICATION
C
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C
C*****
SUBROUTINE LINBOX(AMEAN,BCOV,NSIG,ND)
  DIMENSION AMEAN(4,10),BCOV(4,4,10)
  INTEGER DELLIN
  COMMON/BOX/DELLIN(4,10,2)
  DO (K=1,NSIG)
    : DO (I=1,ND)
    :   : DIF=(SQRT(BCOV(I,I,K))/2.)*3.
    :   : DELLIN(I,K,1)=AMEAN(I,K)-DIF
    :   : DELLIN(I,K,2)=AMEAN(I,K)+DIF
    :   : ...FIN
  : ...FIN
  RETURN
END
```

```

C*****
C
C      LINCLASS
C
C      THIS ROUTINE CLASSIFIES POLYGONS WITH A  LINEAR CLASSIFIER
C      AND CREATES THRESHOLD ARRAY
C
C      SEQUENCE:  LINCLASS INPUT DATAP SIGFIL LINTAP
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS FAUST
C*****
C      DIMENSION THRESH(812), ISW(2)
C      INTEGER DELLIN
C      COMMON/A/ AMEAN(4,10),BCOV(4,4,10),IVX(101),IVY(101),B(400),DET(10)
C      1,LA(133),NVC(10),COUNT(10),IBUF(1650),I1(30),I2(30),I3(30),I4(30)
C      2,IORDER(10),NAM1(10),NAME2(10),NAM3(10),NAM4(10)
C      COMMON/B/JBUF(4,810),IDUN(812)
C      COMMON/BOX/DELLIN(4,10,2)
C      IP=12
C
C      FILE STATEMENTS
C
C      CALL OPEN(1,"COM.CM",1,IERR)
C      CALL COMARG(1,I1,ISW,IERR)
C      CALL COMARG(1,I1,ISW,IERR)
C      CALL COMARG(1,I2,ISW,IERR)
C      CALL COMARG(1,I3,ISW,IERR)
C      CALL COMARG(1,I4,ISW,IERR)
C
C      CALL OPEN(2,I1,0,IE)
C      CALL MTOPD(5,I4,0,IE)
C
C      OPEN TAPE FILE
C
C      CALL MTOPD(3,I2,0,IE)
C      ND=4
C      NUN1=2
C      NUN2=4
C      READ(2,200)NSIG
C      WRITE(IP)NSIG
C
C      GET SIGNATURES  FOR CLASSIFICATION
C
C      CALL GSIG(AMEAN,BCOV,NSIG,NUN1,NUN2,I3,IORDER,NAM1,NAM2,NAM3,NAM4)
C
C      IYMIN=1000
C      CALL LINBOX(AMEAN,BCOV,NSIG,ND)
C      IXMIN=1000
C      IYMAX=0
C      WRITE(IP,202)
C      READ(2,200)NC
C      DO (I=1,NC)
C      :   WRITE(IP,205)I
C      :   READ(2,206)IVX(I),IVY(I)
C      :   WRITE(IP,206)IVX(I),IVY(I)
C      :   IF(IVY(I).LT.IYMIN)IYMIN=IVY(I)
C      :   IF(IVX(I).LT.IXMIN)IXMIN=IVX(I)
C      :   IF(IVY(I).GT.IYMAX)IYMAX=IVY(I)
C      :...FIN
C      IVX(NC+1)=IVX(1)
C      IVY(NC+1)=IVY(1)
C      NV=NC
C      IDEL=IYMAX-IYMIN
C      ISKIP=IYMIN+1

```

SKIP RECORDS

C
C
C
C

```

CALL MTDIO(3,30000K+ISKIP,IBUF,IS,IEE)
DO (LL=1,812)
:   IDUM(LL)=0
...FIN
IDUM(1)=IDEL
IDUM(2)=IXMIN
WRITE(IP) IDUM(1),IDUM(2)
CALL MTDIO(5,50000K+812,IDUM,IS,IER,NW)
WRITE(IP) NW
DO (J=1,IDEL)
:   CALL MTDIO(3,0,IBUF,IS,IE,NW)
:   ISW=2
:   CALL POLY2(ISKIP+J-1,ISW,IVX,IVY,NV,LA)
:   JS=LA(2)
:   JF=LA(3)
:   CALL UNPAC4(IBUF,JBUF,JS,JF,810,1650)
:   JQ=JF-JS+1
:
:
:
:   NN=NSIG
:   CALL LINEAR(JBUF,NN,IC,ND,NN,AMEAN,BCOV,JQ,IDUM,NVG,THRESH)
:
:
:   WRITE(10,207) J
:   JMOVE=JS-1
:   DO (LL=1,810)
:   :   LV=813-LL
:   :   IDUM(LV)=IDUM(LV-2)
:   :   THRESH(LV)=THRESH(LV-2)
:   ...FIN
:   IDUM(1)=JMOVE
:   IDUM(2)=JQ
:   CALL MTDIO(5,50000K+812,IDUM,IS,IER)
:   CALL MTDIO(5,50000K+812,THRESH,IS,IER)
:   TOT=TOT+JQ
:   DO (K=1,JQ)
:   :   LSUB=IDUM(K+2)
:   :   COUNT(LSUB)=COUNT(LSUB)+1
:   ...FIN
...FIN
DO (K=1,NSIG)
:   PER=(COUNT(K)*100)/TOT
:   WRITE(12,102) K,COUNT(K),PER,NAM1(K),NAM2(K),NAM3(K),NAM4(K)
...FIN
CALL MTDIO(5,60000K,IDUM,IS,IER)
CALL MTDIO(5,60000K,IDUM,IS,IER)
102 FORMAT(2X,"CLASS",I4," NUMBER",F7.0," PERCENT",F4.0,2X,4A2)
103 FORMAT(2X,4A2)
200 FORMAT(12)
202 FORMAT(2X,"INPUT # OF CORNERS")
205 FORMAT(2X,"INPUT CORNER #",I3,"J,I")
206 FORMAT(2I4)
207 FORMAT(2X,"LINE #",I5," PROCESSED")
STOP
END

```

C
C
C
C
C
C
C

```

*****
C      LINEAR (SUBROUTINE)
C
C      CLASSIFIES LANDSAT DATA WITH LINEAR DISTANCE
C
C*****
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
SUBROUTINE LINEAR(V,NVM,ICOUNT,ND,NSIG,AMEAN,BCOV,NL,IDUM,NVG,
1THRESH)
  DIMENSION AMEAN(ND,NVM),BCOV(ND,ND,NVM)
  INTEGER DELLIN
  DIMENSION V(ND,NL)
  DIMENSION IDUM(812),THRESH(812),NVG(10)
  INTEGER V
  COMMON/BOX/DELLIN(4,10,2)
  ICOUNT=1
  DO (LL=1,NL) IDUM(LL)=9
  DO (LL=1,NL) THRESH(LL)=0.
  JPT=0
  FOREVER
  :   JPT=JPT+1
  :   IF(JPT.GT.NL) RETURN
  :   IMIN=1
  :   DIST=1.E+6
  :   DO (K=1,NVM)
  :   :   KSUM=0
  :   :   SUM=0.0
  :   :   DO (L=1,ND)
  :   :   :   IF(V(L,JPT).GT.DELLIN(L,K,1).AND.V(L,JPT).LT.DELLIN(L,K,2))
  :   :   :   :   SUM=SUM+ABS(AMEAN(L,K)-V(L,JPT))
  :   :   :   :   KSUM=KSUM+1
  :   :   :   ...FIN
  :   :   ...FIN
  :   :   IF(KSUM.EQ.4)
  :   :   :   IF(SUM.LT.DIST)
  :   :   :   :   DIST=SUM
  :   :   :   :   IMIN=K
  :   :   :   ...FIN
  :   :   ...FIN
  :   :   THRESH(JPT)=DIST
  :   :   IDUM(JPT)=IMIN
  :   :   NVG(IMIN)=NVG(IMIN)+1
  :   :   ...FIN
100  FORMAT(2X,"ENTERED LINEAR ",I6," TIMES")
101  FORMAT(2X,"EXITED LINEAR")
  RETURN
  END

```

```

C*****
C
C      LISTALL
C
C      PRINTS ALL SIGNATURES , VERTICIES , OR HISTOGRAMS
C      FOR A SPECIFIC FILE
C
C      SEQUENCE:  LISTALL FILE
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION IFIL(20),SM(4),COV(4,4),ICOUNT(4,100)
C      DIMENSION IVX(101),IVY(101)
C      DIMENSION ISWS(2),FIELD(17)
C      CALL OPEN(1,"COM.CM",1,IE)
C      CALL COMARG(1,FIELD,ISWS,IE)
C      CALL COMARG(1,FIELD,ISWS,IE)
C      CALL FOPEN(2,FIELD,"B")
C      WRITE(10,555)
555  READ(11) IOUT
C      FORMAT(2X," INPUT OUTPUT DEVICE # (10,12) ")
C      WRITE(10,401)
C      READ(11) ICON
C      WRITE(10,403)
C      READ(11) ITOT
C      READ BINARY(2) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
C      WRITE(10,100) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
C      FOREVER
C      :  READ BINARY(2,END=120) NS1,NS2,NS3,NS4
C      :  IF(ICON.EQ.1) READ BINARY(2) NP,SM,COV
C      :  IF(ICON.EQ.2)
C      :  :  READ BINARY(2) K, IYMIN, IYMAX
C      :  :  DO (K1=1,K) READ BINARY(2) K2, IVX(K1), IVY(K1)
C      :  :  ...FIN
C      :  IF(ICON.EQ.3) READ BINARY(2) ICOUNT
C      :  WRITE(10,101) NS1,NS2,NS3,NS4
C      :  IF(ICON.EQ.1)
C      :  :  IF(ITOT.EQ.1) WRITE(10) NP,SM,COV
C      :  :  ...FIN
C      :  IF(ICON.EQ.2)
C      :  :  IF(ITOT.EQ.1)
C      :  :  :  WRITE(10) K, IYMIN, IYMAX
C      :  :  :  DO (K1=1,K) WRITE(10) K1, IVX(K1), IVY(K1)
C      :  :  :  ...FIN
C      :  :  ...FIN
C      :  IF(ICON.EQ.3)
C      :  :  IF(ITOT.EQ.1) WRITE(10) ICOUNT
C      :  :  ...FIN
C      :  ...FIN
100  FORMAT(1X,6I2)
101  FORMAT(1X,4A2/)
401  FORMAT(2X,"INPUT SWITCH , 1 -SIG , 2 -VER , 3 -HIS")
402  FORMAT(4I10)
403  FORMAT(2X," TOTAL LIST - 1 , OR NAME LIST - 0 ?")
120  STOP
END

```

```

C*****
C
C      LISTSIG
C
C      PRINTS ALL SIGNATURES FOR A GIVEN FILE
C
C      SEQUENCE:  LISTSIG FILE
C*****
C
C      DESIGNED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION IFIL(20),SM(4),COV(4,4),ISW(2)
C      CALL OPEN(1,"COM.CM",1,IERR)
C      CALL COMARG(1,IFIL,ISW,IERR)
C      CALL COMARG(1,IFIL,ISW,IERR)
C      CALL FOPEN(2,IFIL,"B")
C      READ BINARY(2) ITAP1,ITAP2,ITAP3,IDAT1,IDAT2,IDAT3
C      WRITE(12,100) ITAP1,ITAP2,ITAP3,IDAT1,IDAT2,IDAT3
C      FOREVER
C      :   READ BINARY(2,END=120) NS1,NS2,NS3,NS4
C      :   READ BINARY(2) NP,SM,COV
C      :   WRITE(12,101) NS1,NS2,NS3,NS4
C      :   WRITE(12) NP,SM,COV
C      :...FIN
100  FORMAT(1X,6I2)
101  FORMAT(1X,4A2)
120  STOP
      END

```

C-2

ORIGINAL PAGE 1:
OF POOR QUALITY

```
C*****
C
C      MATIN      (SUBROUTINE)
C
C      MATRIX INVERSION PROGRAM
C
C*****
C
C      CREATED AT NASA/JSC      (ASTEP)
C*****
C      SUBROUTINE MATIN(A,N,B,M,KEY,DETERM
C      DIMENSION A(N,N),B(N),IPIVOT(20),INDEX(20,2)
C      DOUBLE PRECISION PIVOT(20), T,SWAP,DETERM,AMAX,ZERO,A,B
C
C      INITIALIZATION
C      5 KEY = N
C      10 DETERM= 1.D+0
C      15 DO 20 J=1,N
C      20 IPIVOT(J)=0
C      30 DO 550 I=1,N
C
C      SEARCH FOR PIVOT ELEMENT
C
C      40 AMAX= 0.D+0
C      45 DO 105 J=1,N
C      50 IF (IPIVOT(J)-1) 60,105,60
C      60 DO 100 K=1,N
C      70 IF (IPIVOT(K)-1) 80,100,740
C      80 IF (DABS(AMAX)-DABS(A(J,K))) 85,100,100
C      85 IROW=J
C      90 ICOLUM=K
C      95 AMAX=A(J,K)
C      100 CONTINUE
C      105 CONTINUE
C      ZERO = 1.D-16
C      107 IF(DABS(AMAX) - ZERO) 745,745,110
C      110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C      INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
C      130 IF (IROW-ICOLUM) 140,260,140
C      140 DETERM=-DETERM
C      150 DO 200 L=1,N
C      160 SWAP=A(IROW,L)
C      170 A(IROW,L)=A(ICOLUM,L)
C      200 A(ICOLUM,L)=SWAP
C      205 IF(M) 260,260,210
C      210 SWAP = B(IROW)
C      230 B(IROW) = B(ICOLUM)
C      250 B(ICOLUM) = SWAP
C      260 INDEX(1,1)=IROW
C      270 INDEX(1,2)=ICOLUM
C      310 PIVOT(I)=A(ICOLUM,ICOLUM)
```



```

C
C
C      DIVIDE PIVOT ROW BY PIVOT ELEMENT
330 A( ICOLUM, ICOLUM) = .1D+1
340 DO 350 L=1,N
350 A( ICOLUM, L) = A( ICOLUM, L) / PIVOT( I)
355 IF( MD 330, 380, 360
360 B( ICOLUM) = B( ICOLUM) / PIVOT( I)
C
C
C      REDUCE NON-PIVOT ROWS
380 DO 550 L1=1,N
390 IF( L1-ICOLUM) 400, 550, 400
400 T= A( L1, ICOLUM)
420 A( L1, ICOLUM) = .0D+0
430 DO 450 L=1,N
450 A( L1, L) = A( L1, L) - A( ICOLUM, L) * T
455 IF( MD 550, 550, 460
460 B( L1) = B( L1) - B( ICOLUM) * T
550 CONTINUE
C
C
C      INTERCHANGE COLUMNS
600 DO 710 I=1,N
610 L=N+1-I
620 IF ( INDEX( L, 1) - INDEX( L, 2) ) 630, 710, 630
630 JROW= INDEX( L, 1)
640 JCOLUM= INDEX( L, 2)
650 DO 705 K=1,N
660 SWAP= A( K, JROW)
670 A( K, JROW) = A( K, JCOLUM)
700 A( K, JCOLUM) = SWAP
705 CONTINUE
710 CONTINUE
    DO 800 I=1,N
    J=N+1-I
800 DETERM= DETERM * PIVOT( J)
740 RETURN
745 DETERM = 0.D+0
746 KEY = I - 1
750 RETURN
    END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      MATPRT
C
C      PRINTS A REAL MATRIX
C
C*****
C
C      CREATED AT NASA/JSC   (ASTEP)
C
C*****
C      SUBROUTINE MATPRT(MAT, NR, RD, NC, NAM)
C      INTEGER RD, P, Q
C      REAL MAT(RD, NC)
1000  FORMAT(1H , 12, 6F10.3)
1010  FORMAT(1H , 6I10)
C      KOUT=12
C      WRITE(KOUT, 1020) NAM, NR, NC
1020  FORMAT(1H , 15X, A2, I6, 3H BY, I3)
C      DO 100 I=1, NR
C      DO 100 J=1, NC
C      IF(MAT(I, J)) 150, 100, 150
100   CONTINUE
C      WRITE(KOUT, 1030)
1030  FORMAT(1H , 15X, 12H ALL ZEROES.)
C      GO TO 30
150   CONTINUE
C      P=0
C      Q=-5
10   P=P+6
C      Q=Q+6
C      IF(NC.LT.P) P=NC
C      WRITE(KOUT, 1010) (J, J=Q, P)
C      WRITE(KOUT, 1000)
C      DO 20 I=1, NR
C      WRITE(KOUT, 1000) I, (MAT(I, J), J=Q, P)
20   CONTINUE
C      IF(NC.GT.P) GO TO 10
30   CONTINUE
C      WRITE(KOUT, 1000)
C      RETURN
C      END
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      MODIFY   (SUBROUTINE)
C
C      PURPOSE
C          COMPUTES WEIGHTED AVERAGE OF TWO MEAN VECTORS
C
C      DESCRIPTION OF PARAMETERS
C
C          INPUT
C              CALLING SEQUENCE
C                  V1 - 1 ST VECTOR MEAN
C                  V2 - 2 ND VECTOR MEAN
C                  N1 - NUMBER OF VECTORS USED TO COMPUTE V1
C                  N2 - NUMBER OF VECTORS USED TO COMPUTE V2
C                  ND - DIMENSION OF V1 AND V2
C
C          OUTPUT
C              CALLING SEQUENCE
C                  V1 - WEIGHTED AVERAGE OF INPUT V1 AND V2
C                  N1 - NUMBER OF VECTORS USED TO COMPUTE OUTPUT V1
C*****
C
C      CREATED AT NASA/JSC   (ASTEP)
C*****
C      SUBROUTINE MODIFY ( V1, V2, N1, N2, ND )
C      DIMENSION V1(ND),V2(ND)
C      REAL N1
C      XN1 = (N1)
C      XN2 = FLOAT(N2)
C      XN1N2 = 1.0/(N1+FLOAT(N2))
C      DO 10 I=1,ND
10  V1(I) = ( XN1*V1(I) + XN2*V2(I) ) * XN1N2
C      N1 = N1 + FLOAT(N2)
C      RETURN
C      END
```

```

*****
C      PACK  (SUBROUTINE)
C
C      PURPOSE
C          PACKS A STORAGE ARRAY TO ELIMINATE A VACATED SLOT, MOVES
C          ALL VECTORS WITH INDEX GREATER THAN INDEX OF VACATED
C          SLOT DOWN ONE POSITION IN THE ARRAY
C
C      DESCRIPTION OF PARAMETERS
C
C          INPUT
C              CALLING SEQUENCE
C                  V - DATA ARRAY
C                  ND - DIMENSION OF EACH VECTOR IN V
C                  NV - NUMBER OF VECTORS IN V
C                  IND - INDEX IN V OF VACATED SLOT
C
C          OUTPUT
C              CALLING SEQUENCE
C                  V - PACKED ARRAY
C
C      *****
C      CREATED AT NASA/JSC  (ASTEP)
C      *****
C      SUBROUTINE PACK(V,ND,NV,IND)
C      DIMENSION V(ND,NV)
C      IF(IND.EQ.NV) RETURN
C      I1=IND+1
C      DO 20 I=I1,NV
C      I2=I-1
C      DO 10 J=1,ND
C10  V(J,I2)=V(J,I)
C20  CONTINUE
C      RETURN
C      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

C*****
C
C      POLY2  (SUBROUTINE)
C
C      CALCULATES POLYGON INTERSECTIONS FOR LINE AND POLYGON
C*****
C
C      CREATED AT NASA/ERL
C
C      PROGRAMMER:  RONNIE PIERSON
C*****
C      SUBROUTINE POLY2(JY, ISW, IX, IY, N, LA)
C      VERTEX -(IX(I), IY(I))=(ELEMENT, SCAN) FOR ITH VERTEX.
C      DIMENSION IX(101), IY(101), D(100), F(101), LA(133), S(101)
C      IF(ISW.EQ.1)GO TO 1
C      SLOPE STORE BLOCK X/Y
C      DO 20 I=1,N
C      IF(IY(I).EQ.IY(I+1))GO TO 20
C      D(I)=FLOAT(IX(I+1)-IX(I))/FLOAT(IY(I+1)-IY(I))
20  CONTINUE
1   Y=FLOAT(JY)
   M=0
   NOV=0
   DO 40 I=1,N
C      LOCATION OF INTERVALS SUCH THAT IY(I).LE.JY.LT.IY(I+1). SCAN=JY
C      IF(IY(I).EQ.JY)GO TO 30
C      IF(IY(I).LT.JY)GO TO 22
C      IF(IY(I+1).LT.JY)GO TO 28
C      GO TO 40
22  IF(IY(I+1).GT.JY)GO TO 28
C      GO TO 40
C      POLYGON BOUNDARY COUNTER AND FUNCTIONAL VALUES. -NON VERTICES-
28  M=M+1
C      F(M)=D(I)*(Y-FLOAT(IY(I)))+FLOAT(IX(I))
C      GO TO 40
C      POLYGON BOUNDARY COUNTER AND FUNCTIONAL VALUES. -VERTICIES-
C      NOV=NUMBER OF VERTICES
30  M=M+1
C      NOV=NOV+1
C      F(M)=FLOAT(IX(I))
C      LA(NOV)=IX(I)
40  CONTINUE
C      K=M
C      BOUNDARY SORT - INCREASING ORDER.
C      DO 50 J=1,K
C      S(J)=F(I)
C      IND=1
C      DO 48 L=1,M
C      IF(F(L).GE.S(J))GO TO 48
C      S(J)=F(L)
C      IND=L
48  CONTINUE
C      F(IND)=10000
50  CONTINUE
C      FOR (NOV.LT.2) INTERVAL S(J),S(J+1) MEMBERSHIP CAN BE DETERMINED
C      IF(NOV.EQ.0)GO TO 500
C      IF(NOV.EQ.1)GO TO 490
C      DATA SHIFT FOR CONSISTENCE AFTER MEMBERSHIP IS DETERMINED
C      DO 52 K=1,M
52  F(K)=S(K)
C      NOI=M-1
C      N=0

```

```

C
C      INTERVAL F(K),F(K+1) MEMBERSHIP SECTION FOR (NOV.GE.2) .
C
DO 200 K=1,NOI
C      A IS THE POINT CHECKED TO DETERMINE INTERVAL F(K),F(K+1) MEMBERSHIP
A=(F(K)+F(K+1))/2.
54      L=0
      J=0
      LV=0
      JV=0
      IN=0
C      DO 122 I=1,N
      LOCATION OF INTERVALS SUCH THAT IX(I).LE.A.LT.IX(I+1)
CX=FLOAT(IX(I))
DX=FLOAT(IX(I+1))
IF(CX.EQ.DX)GO TO 122
IF(CX.EQ.A)GO TO 128
IF(CX.LT.A)GO TO 124
IF(DX.LT.A)GO TO 128
GO TO 122
124      IF(A.GE.DX)GO TO 122
128      CY=FLOAT(IY(I))
      DY=FLOAT(IY(I+1))
      DM=(DY-CY)/(DX-CX)
C      BOUNDARY POINTS (A,F(A)) FOR GIVEN INTERVAL IX(I).LE.A.LT.IX(I+1)
FX=DM*(A-CX)+CY
C      IN=1 INDICATES F(A)=JY
IF(FX.EQ.Y)IN=1
IF(FX.LT.Y)GO TO 130
C      J INDICATES F(A).GT.JY.
C
      J=J+1
      IF(FX.NE.CY)GO TO 122
      IF(A.NE.CX)GO TO 122
C
C      JV INDICATES F(A)=IY(I).GT.JY, FOR SOME I.
C
      JV=JV+1
      GO TO 122
C
C      L INDICATES F(A).LT.JY.
C
130      L=L+1
      IF(FX.NE.CY)GO TO 122
      IF(A.NE.CX)GO TO 122
C
C      LV INDICATES F(A)=IY(I).LT.JY. FOR SOME I.
C
      LV=LV+1
122      CONTINUE

```

```

C
C
C      INTERVAL F(K),F(K+1) MEMBERSHIP DETERMINATION.
C
C      IF(IN.EQ.1)GO TO 150
C      IF(JV.EQ.0)GO TO 140
C      J=L
C      IF(LV.EQ.0)GO TO 140
C
C      IF NO DETERMINATION CAN BE MADE, ADD ANOTHER INCREMENT AND TRY AGA
C
C      A=A+.01
C      GO TO 54
140    L=2*(J/2)
C
C      IF(J.NE.L) THE INTERVAL F(K),F(K+1) IS IN POLYGON.
C
C      IF(J.NE.L)GO TO 150
C
C      CHECK FOR VERTEX AT (F(K);JY)
C
C      DO 146 J=1,NOV
C      CX=FLOAT(LA(J))
C      IF(F(K).NE.CX)GO TO 146
C
C      VERTEX (F(K),JY) INCLUDED.
C
C      M=M+1
C      S(M)=F(K)
C      M=M+1
C      S(M)=F(K)
146    CONTINUE
C      IF(K.NE.NOI)GO TO 200
C      FX=F(K+1)
C      DO 148 J=1,NOV
C      CX=FLOAT(LA(J))
C      IF(FX.NE.CX)GO TO 148
C      M=M+1
C      S(M)=FX
C      M=M+1
C      S(M)=FX
148    CONTINUE
C      GO TO 200
C
C      INTERVAL F(K),F(K+1) INCLUDED
C
150    M=M+1
C      S(M)=F(K)
C      M=M+1
C      S(M)=F(K+1)
200    CONTINUE
C      GO TO 500

```



```

C
C
C
C
C
490  L=2*(M/2)
      IF(L.EQ.M) GO TO 500
      VAV=FLOAT(LA(1))
      M=M+1
      K=M
492  K=K-1
      IF(S(K).EQ.VAV) GO TO 494
      S(K+1)=S(K)
      GO TO 492

C
C
C
494  S(K+1)=VAV

C
C
C
500  IF(M.EQ.0) NO POINTS ON SCAN JY ARE INCLUDED
      IF(M.EQ.0) GO TO 555
      I=0

C
C
C
501  LOOP START AND STOP ROUND OFF TO MIN AND MAX INTEGER VALUES INCLUB
      I=I+1
      LA(I)=S(I)+.999999
      I=I+1
      LA(I)=S(I)
      IF(I.LT.M) GO TO 501
      I=-1

C
C
C
504  INTERVAL(START.GT.STOP) COMPRESS
505  I=I+2
      IF(LA(I).LE.LA(I+1)) GO TO 510
      M=M-2
      IF(M.EQ.0) GO TO 555
      IF(I.GT.M) GO TO 511
      DO 506 K=1,M
506  LA(K)=LA(K+2)
      GO TO 505
510  IF((I+1).LT.M) GO TO 504
511  I=0

C
C
C
      (LOOPSTOP ON INTERVAL(J).EQ.LOOP START ON INTERVAL(J+1)) COMPRESS

512  I=I+2
513  IF(I.EQ.M) GO TO 555
514  IF(LA(I).NE.LA(I+1)) GO TO 512
      M=M-2
      DO 516 K=1,M
516  LA(K)=LA(K+2)
      GO TO 513

C
C
C
      LA(1)=NUMBER OF LOOPS RETURNED

555  IF(M.EQ.0) GO TO 560
      DO 557 I=1,M
      J=M-I+1
557  LA(J+1)=LA(J)
560  LA(1)=M/2
      RETURN
      END

```

```

C*****
C      POLYCL
C      PERFORMS CLUSTERING OPERATION
C      SEQUENCE:  POLYCL INPUT OUTPUT
C*****
C      CREATED AT NASA/JSC    (ASTEP)
C*****
COMMON/UNTNUM/IMGUNT,DATUNT
COMMON/BLANK/KNUM
INTEGER CAR,BUF,ISW(2)
REAL NVG
COMMON/COL/JS,JF
COMMON/BOUND/YMIN,XMIN,YMAX,XMAX
INTEGER YMIN,YMAX,XMIN,XMAX
DIMENSION JPK(101),JPY(101),LA(133)
COMMON/NVEC/NVM
COMMON/BUFFER/BUF(3300)
COMMON/A1/IOUT(812)
INTEGER DATUNT
COMMON/INOUT/NOT,NIN
COMMON/CHARAC/CAR
COMMON/DIST/IDIST
INTEGER PLIST
DIMENSION VM(240),AD(20,20)
DIMENSION NVG(20),PLIST(20),CAR(20),IACOP(4)
DIMENSION RM(20),RV(20)
DIMENSION VMP(60),VAR(60)
DIMENSION ACRE(20),I1(30),I2(30)

C=3.
CALL OPEN(1,"COM.CM",1,IERR)
CALL COMARG(1,I1,ISW,IERR)
CALL COMARG(1,I1,ISW,IERR)
CALL COMARG(1,I2,ISW,IERR)
CALL NTOPD(4,I1,0,IER)
CALL NTOPD(3,I2,0,IER)
RP=3
S=1.
R1=20.
R2=20.
NVMAX=15
NPT=100
NET=500
NMT=100
IP=0
IACOP(1)=2HME
IACOP(2)=2HSI
IACOP(3)=2HAN
IACOP(4)=2HQU
ND=4
NRT=1
IMGUNT=3
DATUNT=4
NOSCAL=1
NBUFSZ=3260
IQ=NBUFSZ/(ND+2)
IBUF2=NBUFSZ-IQ+1
IBUF1=IBUF2-IQ
NIN=11
NOUT=10

```

```

CAR(1)=1HA
CAR(2)=1H.
CAR(3)=1H:
CAR(4)=1H-
CAR(5)=1H,
CAR(6)=1H/
CAR(7)=1H+
CAR(8)=1H0
CAR(9)=1HX
CAR(10)=1H*
CAR(11)=1H$
CAR(12)=1H%
CAR(13)=1HB
CAR(14)=1H@
CAR(15)=1H*

```

USER INPUTS

```

C
C
1
CONTINUE
WRITE(NOUT,1008)
1008 FORMAT(2X,'INPUT VALUES FOR C, RP, R1, R2 AND NVMMAX')
READ(NIN) C, RP, R1, R2, NVMMAX
WRITE(NOUT) C, RP, R1, R2, NVMMAX
WRITE(NOUT,102)
102 FORMAT(22H TYPE YES IF INPUTS OK)
READ(NIN,104) IAN
104 FORMAT(A6)
IF(IAN.NE.3HYES) GO TO 1
IF(NVMMAX.GT.20) NVMMAX=20

```

FIRST PASS INITIALIZATION

```

C
C
C
CALL INITCL(VM,NVG,NVM,ND)
WRITE(10,199)
199 FORMAT(2X,"INPUT 1 FOR POLY , 0 - NORMAL")
READ(11) IPOLY
IF(IPOLY.NE.1) GO TO 201
WRITE(10,202)
202 FORMAT(2X,"INPUT # OF CORNERS")
READ(11) NV
DO 204 I=1,NV
WRITE(10,205) I
205 FORMAT(2X,"INPUT CORNER #",I3," J,I ")
204 READ(11) JPX(I), JPY(I)
JPX(NV+1)=JPX(1)
JPY(NV+1)=JPY(1)
201 CONTINUE
JMIN=10000
IMAX=0
IMIN=10000
DO 405 K1=1,NV
IF(JPX(K1).LT.JMIN) JMIN=JPX(K1)
IF(JPY(K1).LT.IMIN) IMIN=JPY(K1)
405 IF(JPY(K1).GT.IMAX) IMAX=JPY(K1)
NRT=IMAX-IMIN+1
DO 406 K2=1,812
406 IOUT(K2)=0
IOUT(1)=NRT
IOUT(2)=JMIN
LINIT=IMIN
LIN=IMIN+1
CALL MTDIO(3,50000K+812,IOUT,IS1,IER)
CALL MTDIO(4,30000K+LIN,IOUT,IS1,IER)
INRT=0
NEC=0
NMC=0
NPC=0
IPASS=1
JPTP=0
RM(1)=0.
RV(1)=0.
DO 8 I=1,ND
8 VMP(I)=0.
DO 10 I=1,NVM
10 PLIST(I)=1

```

C
C
C

FIRST PASS PROCESSING

20 INRT=INRT+1
IF(INRT.GT.NRT) GO TO 50
IL=LINIT+INRT-1
ISW=IL+1
CALL POLY2(IL, ISW, JPX, JPY, NV, LA)
JS=LA(2)
JF=LA(3)
IF(JS.EQ.0) JS=1
WRITE(NOUT) JS, JF, INRT, "1"
CALL UNPAC1
NX=JF-JS+1
CALL CLUSTA(BUF(1), VM, ND, NX, NVM, NVMMAX, NVG, C, S, RP, R1, NPC,
*NPT, PLIST, NEC, NET, NMIN, NMC, NMT, IPASS, BUF(IBUF1), BUF(IBUF2),
*IP, JPTP, RM, RV, VMP, VAR)
JPTP=JPTP+NX
GO TO 20

ORIGINAL PAGE IS
OF POOR QUALITY

C
C
C

SPECIAL ELIMINATION AND MERGER TESTS

50 NX=0
JPTP=0
NEC=NET
NMC=NMT
CALL CLUSTA(BUF(1), VM, ND, NX, NVM, NVMMAX, NVG, C, S, RP, R1, NPC, NPT,
*PLIST, NEC, NET, NMIN, NMC, NMT, IPASS, BUF(IBUF1),
*BUF(IBUF2), IP, JPTP, RM, RV, VMP, VAR)

C
C
C

SECOND PASS INITIALIZATION

INRT=0
NBACK=LIN+NRT
CALL MTDIO(4, 40000K+NBACK, BUF, IS, IE)
WRITE(10, 222) NBACK
222 FORMAT(2X, " BACKSPACED ", I10, " LINES")
NEC=0
IF(LIN.LE.0) GO TO 444
CALL MTDIO(4, 30000K+LIN, BUF, IS, IE)
444 CONTINUE
NMC=0
NPC=0
IPASS=2
DO 60 I=1, NVM
PLIST(I)=I
60 NVG(I)=0

C
C
C

SECOND PASS PROCESSING

62 INRT=INRT+1
IF(INRT.GT.NRT) GO TO 70
IL=LINIT+INRT-1
ISW=IL+1
CALL POLY2(IL, ISW, JPX, JPY, NV, LA)
JS=LA(2)
JF=LA(3)
JQ=JF-JS+1
DO 402 LL=1, 812
402 IOUT(LL)=0
IOUT(1)=JS
IOUT(2)=JQ
WRITE(NOUT) JS, JF, INRT, "2"
IF(JS.EQ.0) JS=1
CALL UNPAC1
NX=JF-JS+1
CALL CLUSTA(BUF(1), VM, ND, NX, NVM, NVMMAX, NVG, C, S, RP, R2, NPC, NPT,
*PLIST, NEC, NET, NMIN, NMC, NMT, IPASS, BUF(IBUF1),
*BUF(IBUF2), IP, JPTP, RM, RV, VMP, VAR)
JPTP=JPTP+NX
DO 403 LL=1, 810
L2=IBUF2+LL-1
IF(L2.GT.NBUFSZ) GO TO 404
403 IOUT(LL+2)=BUF(L2)
404 CONTINUE
CALL MTDIO(3, 50000K+812, IOUT, IS1, IER)
GO TO 62

PRINT RESULT SUMMARY

```

70 WRITE(NOUT,106)
106 FORMAT(1X,'CLUSTER  SYMBOL  SIZE  R MEAN  R SIGMA  ACRES  ')
   NSAVE=0.0
   DO 72 I=1,NVM
     RV(I)=SQRT(RV(I))
     CC=1.0541515
     IF(NOSCAL.EQ.0) CC=1.53046
     ACRE(I)=NVC(I)*CC
     IF(NVG(I).LT.NSAVE)GO TO 2000
     NSAVE=NVG(I)
     KNUM=I
2000 CONTINUE
72 WRITE(NOUT,108) I,CAR(I),NVG(I),RM(I),RV(I),ACRE(I)
108 FORMAT(16,7X,A1,F10.0,2X,2F7.2,F10.2)
   I=ND*NVM
   DO 74 J=1,I
     VM(J)=VMP(J)
     CALL MTDIO(3,60000K,IOUT,IS1,IER)
     CALL MTDIO(3,60000K,IOUT,IS1,IER)

```

USER OPTION SELECTION

```

76 WRITE(NOUT,110)
110 FORMAT(19H CHOOSE OPTION FROM
WRITE(NOUT,112) IACOP
112 FORMAT(1H,4A8)
   READ(NIN,104) IAN
   IF(IAN.EQ.5HMEANS) GO TO 80
   IF(IAN.EQ.6HSIGMAS) GO TO 85
   IF(IAN.EQ.6HANGDIS) GO TO 90
   IF(IAN.EQ.4HQUIT) STOP
   WRITE(NOUT,114) IAN
114 FORMAT(1H,A6,22H IS NOT A VALID CHOICE)
   GO TO 76

```

MEANS DISPLAY

```

80 CALL MATPRT(VM,ND,ND,NVM,5HMEANS)
   GO TO 76

```

SIGMAS DISPLAY

```

85 I=ND*NVM
   DO 86 J=1,I
     VAR(J)=SQRT(VAR(J))
     CALL MATPRT(VAR,ND,ND,NVM,6HSIGMAS)
     GO TO 76

```

ANGDIS DISPLAY

```

90 IDISF=2
   CALL ANGDIS(VM,NVM,ND,IDISF,AD)
   CALL MATPRT(AD,NVM,NVM,NVM,6HANGDIS)
   GO TO 76
END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

C*****
C
C      PUTIN
C
C      INPUT SIGNATURES
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION COV(4,4),XM(4)
C      CALL OPEN(3,"INSG",0,IE)
C      CALL FOPEN(4,"HSIG","B")
C      READ(3)NSIG
C      ID1=01
C      ID2=22
C      ID3=77
C      IT1=99
C      IT2=99
C      IT3=99
C      WRITE BINARY(4) IT1, IT2, IT3, ID1, ID2, ID3
C      DO (L=1,NSIG)
C      :   WRITE(10,101)
C      :   READ(3,100)NA1,NA2,NA3,NA4
C      :   WRITE(10,102)
C      :   READ(3)NP
C      :   WRITE(10,103)
C      :   READ(3)XM
C      :   DO (K=1,4)
C      :   :   WRITE(10,104)K
C      :   :   READ(3)(COV(K,J),J=1,4)
C      :   :   ...FIN
C      :   WRITE BINARY(4)NA1,NA2,NA3,NA4
C      :   WRITE BINARY(4)NP,XM,COV
C      :   ...FIN
100  FORMAT(4A2)
101  FORMAT(2X,"INPUT NA1,NA2,NA3,NA4")
102  FORMAT(2X,"INPUT NUMBER OF POINTS")
103  FORMAT(2X,"INPUT (XM(I),I=1,4)")
104  FORMAT(2X,"INPUT ROW # ",I1," OF COV - (COV(K,J),J=1,4)")
      STOP
      END

```

```

C*****
C
C      RAINBOW
C
C      DISPOSES EXISTING PSEUDOCOLOR MEMORY TO A DISK FILE
C      DISPOSES PSEUDOCOLOR MEMORY IN DISK FILE TO DISPLAY
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  FRED L. THOMPSON
C*****
C      INTEGER IORAN(94)
C      CALL FOPEN (5,"RYGBV.")
C      ACCEPT "TYPE 1 TO SAVE PRESENT PSEUDOCOLOR ", IANS1
C      IF (IANS1.EQ.1)
C      :   REWIND 5
10  :   DO (J=1,1000) READ BINARY (5,END=10) IORAN
C      :   DO (I=1,94) IORAN(I)=0
C      :   CALL RCM (0, IORAN)
C      :   TYPE "TYPE A 30 CHARACTER DISCREPTION"
C      :   TYPE "....."
20  :   READ (11,20) (IORAN(I), I=65,94)
C      :   FORMAT (30A1)
C      :   WRITE BINARY (5) IORAN
C      :   TYPE "PSEUDOCOLOR MEMORY NUMBER ",J
C      ...FIN
C      ACCEPT "TYPE 1 TO REPLACE PRESENT PSEUDOCOLOR ", IANS2
C      IF (IANS2.EQ.1)
C      :   ACCEPT "TYPE 1 IF YOU KNOW THE PSEUDOCOLOR MEMORY NUMBER ", IANS3
C      :   IF (IANS3.NE.1)
30  :   :   PAUSE PUT UP PSEUDOCOLOR TEST PATTERN HIT RETURN
C      :   :   REWIND 5
C      :   :   DO (J=0,1000)
C      :   :   :   READ BINARY (5,END=30) IORAN
C      :   :   :   WRITE (10,40) (IORAN(I), I=65,94)
40  :   :   :   FORMAT (1X,30A1)
C      :   :   :   CALL WCM (0, IORAN)
C      :   :   :   ACCEPT "TYPE 1 FOR NEXT PATTERN ", IANS4
C      :   :   :   IF (IANS4.NE.1) STOP NORMAL EXIT
C      :   :   ...FIN
C      :   ...FIN
C      :   IF (IANS3.EQ.1)
C      :   :   REWIND 5
C      :   :   ACCEPT "PSEUDOCOLOR MEMORY NUMBER ? ", INUM
C      :   :   DO (J=1, INUM) READ BINARY (5) IORAN
C      :   :   :   WRITE (10,60) (IORAN(I), I=65,94)
60  :   :   :   FORMAT (1X,30A1)
C      :   :   :   CALL WCM (0, IORAN)
C      :   :   ...FIN
C      ...FIN
C      STOP NORMAL EXIT
C      END

```

```

C*****
C
C      RATIO
C
C      THIS PROGRAM DIVIDES ONE ERTS CHANNEL INTO THE OTHER
C      THREE TO LEAVE THREE CHANNELS OF DATA. IT ALSO WILL TAKE
C      TWO CHANNELS AND BY DIVIDING ONE INTO THE OTHER TWO, CREATE
C      A THIRD CHANNEL, ALL TO BE DISPLAYED.
C
C      SEQUENCE:  RATIO MTU:F
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  MICHAEL D. FURMAN
C*****
C      INTEGER INPUT(1700), IMP(3400), IMAG(0:255), IMAG2(0:255), ITAPE(10)
C      INTEGER ISW(2)
C      CALL OPEN(1,"COM.CM",1,IERR)
C      CALL COMARG(1,ITAPE,ISW,IERR)
C      CALL COMARG(1,ITAPE,ISW,IERR)
C      CALL MTOPD(2,ITAPE,0,IE)
C      ACCEPT "TWO (2) OR FOUR (4) CHANNEL RATIO ",IANS1
C      WHEN (IANS1.EQ.2) ACCEPT "INPUT NUM. AND DENOM. CHANNELS ",INUM,IDEN
C      ELSE ACCEPT "INPUT DENOMINATOR CHANNEL ",IDEN
C      ACCEPT "SCALE FACTOR ",ISCAL
C      ACCEPT "INPUT LINE AND ELEMENT ",LINE,N
C      IF (((N/2)*2).EQ.N) N=N-1
C      ISKP=1+LINE
C      IEL=(N*4)-(6/(1+N-((N/2)*2)))-1
C      IEND=IEL+1019
C      IDEN2=IDEN*2-1
C      DO (I=1,ISKP) CALL MTDIO(2,0,INPUT,IS,IE,IC)
C      DO (K2=1,255)
C      :   CALL MTDIO(2,0,INPUT,IS,IE,IC)
C      :   CALL UPAC8(INPUT,IMP,1700)
C      :   M2=0
C      :   M1=0
C      :   DO (K=1,7,2)
C      :   :   L=0
C      :   :   IF (K.NE.IDEN2)
C      :   :   :   DO (J1=IEL,IEND,8)
C      :   :   :   :   IMAG(L)=(IMP(J1+K)/(IMP(IDEN2+J1)+1.0))*ISCAL
C      :   :   :   :   IMAG(L+1)=(IMP(J1+K+1)/(IMP(IDEN2+J1+1)+1.0))*ISCAL
C      :   :   :   :   L=L+2
C      :   :   :   ...FIN
C      :   :   :   DO (I=0,255)
C      :   :   :   :   IF (IMAG(I).GT.255)
C      :   :   :   :   :   TYPE "ATTEMPTED SCALE TO ",IMAG(I)
C      :   :   :   :   :   IMAG(I)=255
C      :   :   :   :   ...FIN
C      :   :   :   ...FIN
C      :   :   CALL PAC8(IMAG,IMAG2,256)
C      :   :   CALL RIMWRITE(M1,M2,IMAG2,256)
C      :   :   WHEN (M2.EQ.0) M2=255
C      :   :   ELSE M2=0
C      :   :   M1=M1+1
C      :   :   ...FIN
C      :   ...FIN
C      ...FIN
C      STOP
C      END

```



```

C*****
C
C      RDATA
C
C      THIS SUBROUTINE UNPACKS DATA FOR A POLYGON
C
C*****
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
SUBROUTINE RDATA( ITOP, IBOT, NV, IX, IY, IXD, NP)
  DIMENSION LA(133), IX(101), IY(101), JBUF(100,4), IXD(512,4)
  COMMON /BUFFER/ IBUF(1700)
  COMMON /INDEV/ IPOS
  IC=0
  ISUM1=ITOP-IPOS-1
222  FORMAT(2X, "SKIPPED ", I5, " RECORDS")
  IF( ISUM1.GT.0) CALL MTDIO(2,30000K+ISUM1, IBUF, IS, IER)
  IF( ISUM1.LT.0) CALL MTDIO(2,40000K-ISUM1, IBUF, IS, IER)
  IDEL=IBOT-ITOP+1
  DO  ( IL=1, IDEL)
    :  CALL MTDIO(2,0, IBUF, IS, IE)
    :  ISW=ITOP+1
    :  ILIN=ITOP+IL-1
    :  CALL POLY2( ILIN, ISW, IX, IY, NV, LA)
    :  NN=100
    :  NSEC=1
    :  MM=1650
    :  DO  (K=1, NSEC)
    :    :  ISUB1=2*K
    :    :  JS=LA( ISUB1)
    :    :  JF=LA( ISUB1+1)
    :    :  JD=JF-JS+1
    :    :  CALL UNPAC3( IBUF, JBUF, JS, JF, NN, MM)
    :    :  DO  (K=1, JD)
    :    :    :  IC=IC+1
    :    :    :  DO  (KD=1,4)
    :    :    :    :  IXD( IC, KD)=JBUF( K, KD)
    :    :    :  ...FIN
    :    :  ...FIN
    :  ...FIN
  ...FIN
  NP=IC
  IPOS=IBOT
  WRITE( 10, 222) IDEL
  TYPE "  TOTAL # PTS = ", NP
  RETURN
END

```

```

C*****
C
C      RDIMG  (SUBROUTINE)
C
C      READ AND DISPLAY IMAGE TO SCREEN
C
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  ROBERT A. MADDOX
C*****
SUBROUTINE RDIMG(IL,IEL,NCH,IBLUP)
  INTEGER IMAG(260),IMAG2(0:512),NCH(0:2)
  COMMON/BUFFER/INPUT(1700),/INDEV/IP0S
  ISUM1=IL-IP0S-1
  IELP=((IEL+254)/2)*4
  IEL2=((IEL-1)/2)*4
  IF (IBLUP.LE.0)
    :   TYPE "ILLEGAL FACTOR",IBLUP
    :   RETURN
  ...FIN
  IF (ISUM1.GT.0) CALL MTDIO(2,30000K+ISUM1,INPUT,IS,IE,IC)
  IF (ISUM1.LT.0) CALL MTDIO(2,40000K-ISUM1,INPUT,IS,IE,IC)
  NCH(0)=1
  ICC=0
  NCH(1)=2
  NCH(2)=4
  M7=255/IBLUP
  DO (K1=0,M7)
    :   CALL MTDIO(2,0,INPUT,IS,IE,IC)
    :   DO (I5=0,2)
    :     :   M5=0
    :     :   IF (I5.EQ.1) M5=255
    :     :   I2=NCH(I5)
    :     :   I3=IEL2+I2
    :     :   I4=IELP+I2
    :     :   L=1
    :     :   DO (J=I3,I4,4)
    :     :     :   JJ=J
    :     :     :   IF(JJ.GT.1700)JJ=1700
    :     :     :   IMAG(L)=INPUT(JJ)
    :     :     :   L=L+1
    :     :     :   ...FIN
    :     :   IF (IBLUP.GT.1) BLOW-UP-IMAGE
    :     :   DO (I=1,IBLUP)
    :     :     :   ICC=ICC+1
    :     :     :   IF(ICC.GT.768)GO TO 100
    :     :     :   CALL RINWR(I5,M5,IMAG,256)
    :     :     :   ...FIN
    :     :   ...FIN
  ...FIN
100 CONTINUE
  IP0S=IL+M7
  RETURN
  TO BLOW-UP-IMAGE
  :   CALL UPAC8(IMAG,IMAG2,256)
  :   DO (K7=0,255)
  :     :   K9=256-K7
  :     :   KDB=K9/IBLUP
  :     :   IMAG2(K9)=IMAG2(KDB)
  :     :   ...FIN
  :   CALL PAC8(IMAG2,IMAG,512)
  :   ...FIN
END

```

```

C*****
C
C      RETING   (SUBROUTINE)
C
C      PURPOSE
C          RETRIEVES IMAGE AND THRESHOLD ARRAYS CORRESPONDING TO
C          A DATA RECORD
C
C      DESCRIPTION OF PARAMETERS
C
C          INPUT
C              CALLING SEQUENCE
C                  IFLAG - FIRST ENTRY OR REWIND AND SKIP HEADING RECORD
C                      FLAG
C                  NUNIT - UNIT NUMBER
C
C          OUTPUT
C              CALLING SEQUENCE
C                  IMG - IMAGE ARRAY
C                  THR - CORRESPONDING THRESHOLD ARRAY
C                  NP - NUMBER OF POINTS IN EACH ARRAY
C
C*****
C
C      CREATED AT NASA/JSC   (ASTEP)
C
C*****
C      SUBROUTINE RETING( IFLAG,NUNIT, IMG, THR, NP)
C      DIMENSION IMG(112)
C      COMMON/COL/JS,JF
C      INTEGER THR
C      IF( IFLAG.NE.1) GO TO 10
C      REWIND NUNIT
C      READ BINARY(NUNIT) JS,NP,( IMG( I) , I=1, NP)
C      RETURN
C      END
10

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      SAVING  (SUBROUTINE)
C
C      PURPOSE
C          SAVES, ON TAPE, IMAGE AND THRESHOLD ARRAYS CORRESPONDING
C          TO A DATA RECORD
C
C      DESCRIPTION OF PARAMETERS
C
C          INPUT
C              CALLING SEQUENCE
C                  IFLAG - FIRST ENTRY OR REWIND AND WRITE HEADING RECORD
C                      FLAG
C                  NUNIT - UNIT NUMBER
C                  IMG - IMAGE ARRAY TO BE SAVED
C                  THR - THRESHOLD ARRAY CORRESPONDING TO IMG
C                  NP - NUMBER OF POINTS IN EACH ARRAY
C*****
C
C      CREATED AT NASA/JSC  (ASTEP)
C*****
C      SUBROUTINE SAVING( IFLAG, NUNIT, IMG, THR, NP)
C      DIMENSION IMG(NP), THR(NP)
C      INTEGER THR
C      DIMENSION BLK(166)
C      COMMON/COL/JS, JF
C      BLK(1)=1.
C      IF( IFLAG.NE.1) GO TO 10
C      REWIND NUNIT
C10  WRITE BINARY(NUNIT) JS, NP, ( IMG(K), K=1, NP)
C      WRITE(12) NP, IMG(1)
C      RETURN
C      END
```

SCORECARD

THIS ROUTINE CLASSIFIES TRAINING FIELDS WITH A
MAXLIK CLASSIFIER

SEQUENCE: SCORECARD INPUT TAPNAM SIGFIL VERTEX

CREATED AT GEORGIA TECH EES

PROGRAMMER: NICHOLAS L. FAUST

DIMENSION AMEAN(4,20),BCOV(4,4,20),IVX(101),IVY(101),B(400),DET(20)
1,LA(133),NVC(20),ICOUNT(20),IBUF(1650),I1(30),I2(30),I3(30),I4(30)
2,IORDER(20),NAM1(20),NAM2(20),NAM3(20),NAM4(20),ISW(2)
COMMON/DTRANS/IDAT(256),JBUF(512,4),IDUM(512)
IP=12

FIELD AND OPEN STATEMENTS

CALL OPEN(1,"COM.CM",1,IERR)
CALL COMARG(1,I1,ISW,IERR)
CALL COMARG(1,I1,ISW,IERR)
CALL COMARG(1,I2,ISW,IERR)
CALL COMARG(1,I3,ISW,IERR)
CALL COMARG(1,I4,ISW,IERR)

CALL OPEN(2,I1,0,IE)
CALL FOPEN(7,"MAXFLD","B")

OPEN TAPE FILE

CALL MTOPD(3,12,0,IE)
ND=4
NUN1=2
NUN2=4
READ(2,200)NSIG
WRITE(IP)NSIG

GET SIGNATURES FOR CLASSIFICATION

CALL GSIG(AMEAN,BCOV,NSIG,NUN1,NUN2,I3,IORDER,NAM1,NAM2,NAM3,NAM4)
CALL ICD(AMEAN,BCOV,ND,NSIG,DET,B,B)

ISIG=0

```

DO (K=1, NSIG)
: ISIG=ISIG+1
: CALL FOPEN(5, 14, "B")
: READ BINARY(5) ITAP1, ITAP2, ITAP3, IDAT1, IDAT2, IDAT3
: K3=IORDER(KSIC)
: DO (K2=1, K3)
: : READ BINARY(5) NA1, NA2, NA3, NA4
: : READ BINARY(5) K, IYMIN, IYMAX
: : DO (I=1, K)
: : : READ BINARY(5) K1, IVX(I), IVY(I)
: : : ...FIN
: : ...FIN
: CALL FCLOSE(5)
: WRITE(IP) ITAP1, ITAP2, ITAP3
: WRITE(IP) IDAT1, IDAT2, IDAT3
: WRITE(IP, 103) NA1, NA2, NA3, NA4
: WRITE(IP) K, IYMIN, IYMAX
: DO (L3=1, K)
: : WRITE(IP) K1, IVX(L3), IVY(L3)
: : ...FIN
: NV=K
: ISKIP=IYMIN+1
: IVX(K+1)=IVX(1)
: IVY(K+1)=IVY(1)
: IDEL=IYMAX-IYMIN+1
: DO (II=1, 20)
: : ICOUNT(II)=0
: : ...FIN
: JTOT=0

C
C
C
: /*SKIP RECORDS
: CALL MTDIO(3, 30000K+ISKIP, IBUF, IS, IEE)
: WRITE BINARY(7) IDEL, ISIG
: DO (J=1, IDEL)
: : CALL MTDIO(3, 0, IBUF, IS, IE, NW)
: : ISW=2
: : CALL POLY2(ISKIP+J-1, ISW, IVX, IVY, NV, LA)
: : JS=LA(2)
: : JF=LA(3)
: : CALL UNPAC3(IBUF, JBUF, JS, JF, 512, 1650)
: : JQ=JF-JS+1
: : WRITE(IP) J, JS, JF, JQ, IDEL, ISKIP
: : DO (L=1, JQ)
: : : DO (L1=1, ND)
: : : : ISUB=(L-1)*ND+L1
: : : : IBUF(ISUB)=JBUF(L, L1)
: : : : ...FIN
: : : ...FIN
: : :
: : CALL MAXLI( IBUF, ND, JQ, AMEAN, BCOV, NSIG, DET, IDUM, NVG)
: :
: : JMOVE=JS-1
: : WRITE BINARY(7) JMOVE, JQ, IDUM
: : JTOT=JTOT+JQ
: : DO (K=1, JQ)
: : : LSUB=IDUM(K)
: : : ICOUNT(LSUB)=ICOUNT(LSUB)+1
: : : ...FIN
: : ...FIN
: DO (K=1, NSIG)
: : JPER=(ICOUNT(K)*100)/JTOT
: : WRITE(12, 102) K, ICOUNT(K), JPER, NAM1(K), NAM2(K), NAM3(K), NAM4(K)
: : ...FIN
: ISUM=IDEL+ISKIP
: CALL MTDIO(3, 40000K+ISUM, IBUF, IS, IER)
: ...FIN
102 FORMAT(2X, "CLASS ", 14, " NUMBER ", 16, " PERCENT ", 13, 2X, 4A2)
103 FORMAT(2X, 4A2)
200 FORMAT(12)
STOP
END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

C*****
C
C      SEQCOV   (SUBROUTINE)
C
C      PURPOSE
C
C      RECURSIVE CALCULATION OF SAMPLE COVARIANCE MATRICIES AND
C      MEAN VECTORS
C
C      INPUT
C
C          CALLING SEQUENCE
C          X = CURRENT SAMPLE VECTOR
C          N = DIMENSION OF X
C          V = PREVIOUS COVARIANCE MATRIX
C          M = PREVIOUS MEAN VECTOR
C          I = PREVIOUS # OF SAMPLES
C          D = SCRATCH AREA
C
C      OUTPUT
C
C          V = CURRENT COVARIANCE MATRIX
C          M = CURRENT MEAN VECTOR
C          I = CURRENT # OF SAMPLES
C*****
C
C      CREATED AT NASA/JSC   (ASTEP)
C*****
C      SUBROUTINE SEQCOV(X,N,V,M,I,D)
C      INTEGER X
C      DIMENSION X(N)
C      DIMENSION V(N,N)
C      REAL M(N)
C      DIMENSION D(N,2)
C      WHEN(I.LE.0)
C      :   DO (L=1,N)
C      :   :   V(K,L)=0.0
C      :   :   I=1
C      :   :   RETURN
C      :   ...FIN
C      ...FIN
C      ELSE
C      :   FI=FLOAT(I)
C      :   FIP=1./FI
C      :   I=I+1
C      :   FI=FLOAT(I)
C      :   FIP=1./FI
C      :   DO (K=1,N)
C      :   :   D(K,1)=X(K)-M(K)
C      :   :   D(K,2)=D(K,1)*FI
C      :   :   M(K)=M(K)+D(K,2)
C      :   ...FIN
C      :   FIP=1.-FIP
C      :   DO (K=1,N)
C      :   :   DO (L=1,N)
C      :   :   :   V(K,L)=FIP*V(K,L)+D(K,2)*D(L,1)
C      :   :   :   ...FIN
C      :   :   ...FIN
C      :   ...FIN
C      RETURN
C      END

```

```

C*****
C
C      SEQST  (SUBROUTINE)
C
C      PURPOSE      SEQUENTIALLY CALCULATED MEAN VECTOR AND VARIANCES
C
C      DESCRIPTION OF PARAMETERS
C
C      INPUT
C      CALLING SEQUENCE
C      XM - CURRENT MEAN
C      XV - CURRENT VARIANCES
C      N - NUMBER OF POINTS USED TO COMPUTE XM AND XV
C      ND - DIMENSION OF DATA VECTOR
C      X - NEW DATA VECTOR TO BE ADDED TO XM AND XV
C
C      OUTPUT
C      CALLING SEQUENCE
C      XM,XV,N - UPDATED VALUES
C*****
C
C      CREATED AT NASA/JSC  (ASTEP)
C*****
C      SUBROUTINE SEQST(XM,XV,N,ND,X)
C      DIMENSION XM(ND),XV(ND),X(ND)
C      INTEGER X
C      IF(N.GT.0) GO TO 20
C      DO 10 I=1,ND
C      XM(I)=X(I)
C 10  XV(I)=0.
C      N=1
C      RETURN
C 20  F1=(1.-1./FLOAT(N))
C      N=N+1
C      F2=1./FLOAT(N)
C      DO 30 I=1,ND
C      T=X(I)-XM(I)
C      XM(I)=XM(I)+F2*T
C 30  XV(I)=F1*XV(I)+F2*(T**2)
C      RETURN
C      END

```


STRIPLESS

PROGRAM TO DESTRIPE A DATA SET GIVEN THE
CONVERSION TABLE FROM STRNF2

SEQUENCE: STRIPLESS MTIN:F MTOUT:F TABLE

PROGRAMMERS: RONNIE PIERSON (NASA/ERL)
NICKOLAS FAUST (GEORGIA TECH EES)

```

DIMENSION IEM(4000),LT(4,6,128),IZ(2,4,412)
DIMENSION ITAP(15),JTAP(15),IFIL(15),ISW(2)
CALL OPEN(1,"COM.CM",1,IERR)
CALL COMARG(1,ITAP,ISW,IERR)
CALL COMARG(1,ITAP,ISW,IERR)
CALL COMARG(1,JTAP,ISW,IERR)
CALL COMARG(1,IFIL,ISW,IERR)
CALL FOPEN(3,IFIL,"E")
CALL MTOPD(2,ITAP,0,IERR)
CALL MTOPD(4,JTAP,0,IERR)
NOFT1=1
READ BINARY(3)LT
DO (K=1,2)
: CALL MTDIO(2,0,IEM,IS,IERR,NW)
: CALL MTDIO(4,50000K+NW,IEM,IS,IERR)
...FIN
N=0
NL=0
TYPE "HOW MANY LINES OF 6 TO SKIP?"
READ(11)KICK
TYPE "HOW MANY LINES TO PROCESS?"
READ(11)NLINE
KICK6=6*KICK
IF(KICK.GT.0)
: CALL MTDIO(2,30000K+KICK6,IEM,IS,IERR)
...FIN
FOREVER
: CALL MTDIO(2,0,IEM,IS,IERR,NW)
: N=N+1
: NL=N+1
: IF(NL.GT.NLINE)WRITE-EOF
: IF(NW.LE.50)WRITE-EOF
: CALL PACKER(1,IEM,IZ)
: DO (I=1,4)
: : DO (J=1,407)
: : : K=IZ(1,I,J)+1
: : : K1=IZ(2,I,J)+1
: : : IF(K.GT.128)K=128
: : : IF(K1.GT.128)K1=128
: : : IZ(2,I,J)=LT(1,N,K1)
: : : IZ(1,I,J)=LT(1,N,K1)
: : ...FIN
: ...FIN
: CALL PACKER(2,IEM,IZ)
: CALL MTDIO(4,50000K+1650,IEM,IS,IERR)
: IF(N.EQ.6)N=0
...FIN
TO WRITE-EOF
: CALL MTDIO(4,60000K,IEM,IS,IERR)
: CALL MTDIO(4,60000K,IEM,IS,IERR)
: CALL MTDIO(4,10000K,IEM,IS,IERR)
: CALL MTDIO(2,10000K,IEM,IS,IERR)
: STOP
...FIN
END

```

```

C*****
C
C      STRIPNF
C
C      TO ELIMINATE STRIPING IN LANDSAT DATA
C
C      SEQUENCE:  STRNF2 TAPE TABFIL
C*****
C
C      CREATED AT NASA/ERL
C
C      PROGRAMMER:  RON PIERSON
C*****
      DIMENSION IER(620,8),LT(4,6,128),IUNPAK(2,4,3),DT(4,8,129)
      DIMENSION IEM(1650),LIMS(4,6,2)
      DIMENSION ITAP1(15),ITAB(15)
      COMMON/DUM/IER,DT,LIMS,IUNPAK,ITAP1
      EQUIVALENCE (IEM(1),IER(1,6))
      EQUIVALENCE (LT(1,1,1),IER(1,1))
      CALL FIELD(2,ITAP1,$701)
      CALL FIELD(3,ITAB,$701)
      CALL MTOPD(1,ITAP1,0,IERR)
      CALL FOPEN(3,ITAB,"B")
      NOFTR=0
      NL1=0
      NOFT1=1
      NOFT=1
      WRITE(10,2222)
2222  FORMAT(2X,"INPUT NPRNT NLINES")
      READ(11)NPRNT,NLINES
99    CALL MTDIO(1,0,IEM,IS,IERR)
      CALL MTDIO(1,0,IEM,IS,IERR)
C      SIX SCAN READ
10    N=1
      IERROR=0
11    CALL MTDIO(1,0,IEM,IS,IERR,NW)
      IF(NW.LT.1600)GO TO 400
      IF(NL1.GT.NLINES)GO TO 400
      DO (KV=1,620) IER(KV,N)=IEM(KV)
      N=N+1
      NL1=NL1+1
      IF(N.LT.7)GO TO 11
C      SIX BY SIX PROCESS
      N=1
      NUM=0
      DO 299 NE=1,51
      N=N+12
      DO 280 J=1,6
      DO (L1=1,3)
      :   DO (L2=1,4)
      :   :   N1=N+(L1-1)*4+L2-1
      :   :   IWRD=IER(N1,J)
      :   :   IUNPAK(1,L2,L1)=ISHFT(IWRD,-8)
      :   :   IUNPAK(2,L2,L1)=IAND(IWRD,377K)
      :   :   ...FIN
      :   ...FIN
C      ELEMENT AVERAGE
      DO 280 I=1,4
      DO 200 K1=1,3
      NUM=NUM+IUNPAK(1,I,K1)+IUNPAK(2,I,K1)
200  CONTINUE
      DT(I,J,1)=FLOAT(NUM)/6.
      NUM=0
280  CONTINUE
      DO 284 I=1,4
C      DISCARD OF ABNORMAL DATA VALUES
      SM=DT(I,1,1)
      BG=SM

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

DO 279 J=2,6
RAD=DT(I,J,1)
IF(RAD.GT.SM)GO TO 275
SM=RAD
GO TO 279
275 IF(RAD.LT.BG)GO TO 279
BG=RAD
279 CONTINUE
RAD=BG-SM
IF(RAD.GT.10.)GO TO 284
C DATA AGREGATION
AVG=0.
DO 282 J=1,6
282 AVG=DT(I,J,1)+AVG
MN=AVG/6.+5
IF(MN.LT.0)GO TO 284
IF(MN.GT.127)GO TO 284
MN=MN+2
D=DT(1,7,MN)+6.
DO 283 J=1,6
283 DT(I,J,MN)=(DT(I,J,MN)*DT(1,7,MN)+6.*DT(I,J,1))/D
CONTINUE
C POINT TOTAL UPDATE
DT(1,7,MN)=D
284 CONTINUE
299 CONTINUE
GO TO 10
400 NOFTR=NOFTR+1
IF(NOFTR.LT.NOFT)GO TO 99
C DETERMINATION OF USEABLE DATA RANGES
DO 1099 I=1,4
J=2
1013 J=J+1
IF(DT(1,7,J).LT.299.)GO TO 1013
IER(I,6)=J
J=125
1017 J=J-1
IF(DT(1,7,J).LT.299.)GO TO 1017
1099 IER(I+4,6)=J
C MEAN ADJUST
DO 410 I=1,4
L1=IER(I,6)
K1=IER(I+4,6)
DO 410 NE=L1,K1
SUM=0.
DO 405 J=1,6
405 SUM=SUM+DT(I,J,NE)
AVG=SUM/6.
A=0.
L=1
K=3
DO 406 J=1,6
B=DT(I,J,NE)-AVG
IF(B.LT.0)B=-B
IF(B.LT.A)GO TO 406
A=B
L=J
406 CONTINUE
SUM=SUM-DT(I,L,NE)
AVG=SUM/5.
IF(L.EQ.K)K=1
A=0.
DO 407 J=1,6
IF(J.EQ.L)GO TO 407
B=DT(I,J,NE)-AVG
IF(B.LT.0.)B=-B
IF(B.LT.A)GO TO 407
A=B
K=J
407 CONTINUE

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

SUM=SUM-DT(I,K,NE)
410 DT(I,8,NE)=SUM/4.
C SMOOTHING OF SPARSE DATA OVER USEABLE DATA RANGES
DO 1199 I=1,4
J=IER(I,6)+1
K=IER(I+4,6)-1
DO 1199 L=J,K
IF(DT(I,7,L).GT.299.)GO TO 1199
N=L
1113 N=N-1
IF(DT(I,7,N).LT.299.)GO TO 1113
M=L
1117 M=M+1
IF(DT(I,7,M).LT.299.)GO TO 1117
RAD=DT(I,7,L)/300.
SM=1.-RAD
IF(DT(I,7,L).LT.10.)DT(I,8,L)=FLOAT(L-2)
A=(DT(I,8,L)-DT(I,8,N))/(DT(I,8,M)-DT(I,8,N))
DO 1121 L1=1,6
BG=DT(I,L1,M)-DT(I,L1,N)
BG=A*BG+DT(I,L1,N)
1121 DT(I,L1,L)=SM*BG+RAD*DT(I,L1,L)
1199 CONTINUE
IF(NPRNT.EQ.0)GO TO 500
DO 499 NE=2,63
MN=NE-2
WRITE(12,401)MN
401 FORMAT(9X,"BAND 1",6X,"BAND 2",6X,"BAND 3",6X,"BAND 4",6X,"MEAN",12)
DO 404 J=1,8
A=DT(1,J,NE)
B=DT(2,J,NE)
C=DT(3,J,NE)
D=DT(4,J,NE)
WRITE(12,402)J,A,B,C,D
402 FORMAT(2X,"DET",1X,11,4(3X,F9.2))
404 CONTINUE
499 CONTINUE
C MID-RANGE TABLE BUILDER
500 DO 498 I=1,4
DO 498 NE=1,6
L=IER(I,6)
K=IER(I+4,6)
SM=DT(I,NE,L)
K=DT(I,NE,K)
J=SM
A=FLOAT(J)
IF(SM.NE.A)J=J+1
LIMS(I,NE,1)=J
LIMS(I,NE,2)=K+2
DO 498 M=J,K
A=FLOAT(M)
N=M+1
431 IF(DT(I,NE,L).LE.A)GO TO 435
L=L-1
GO TO 431
435 L2=L+1
IF(DT(I,NE,L2).GT.A)GO TO 441
L=L+1
GO TO 431
441 A2=DT(I,NE,L2)-DT(I,NE,L)
B=A-DT(I,NE,L)
B2=DT(I,8,L2)-DT(I,8,L)
LT(I,NE,N)=DT(I,8,L)+B*B2/A2+.5
498 L=L+1

```

```

C      BELOW-RANGE TABLE BUILDER
DO 415 I=1,4
DO 415 NE=1,6
K=LIMS(I,NE,1)
L=K
DO 415 J=1,K
M=LT(I,NE,L+1)-1
IF(M.LT.0)M=0
LT(I,NE,L)=M
415 L=L-1
C      ABOVE-RANGE TABLE BUILDER
DO 425 I=1,4
DO 425 NE=1,6
K=LIMS(I,NE,2)
DO 425 L=K,128
M=LT(I,NE,L-1)+1
IF(M.GT.127)M=127
425 LT(I,NE,L)=M
C      TABLE SMOOTHER
DO 525 I=1,4
DO 525 J=1,6
DO 525 K=2,126
M=K+1
L=LT(I,J,M)-1
IF(L.LE.LT(I,J,K))GO TO 525
N=K-1
IF(LT(I,J,N).LT.LT(I,J,K))GO TO 521
LT(I,J,K)=LT(I,J,K+1)
GO TO 525
521 L=K+2
IF(LT(I,J,L).GT.LT(I,J,M))GO TO 525
LT(I,J,M)=LT(I,J,M-1)
525 CONTINUE
DO 541 I=1,4
DO 541 J=1,6
541 LT(I,J,1)=0
IF(NPRNT.EQ.0)GO TO 550
DO 549 K=1,128
MN=K-1
WRITE(12,601)MN
601 FORMAT(6X,"CHNL 1  CHNL 2  CHNL 3  CHNL 4  INPUT=",I3)
DO 549 J=1,6
L=LT(1,J,K)
L2=LT(2,J,K)
N=LT(3,J,K)
N=LT(4,J,K)
WRITE(12,603)J,L,L2,M,N
603 FORMAT(2X," DET ",I1,3X,4(I3,6X))
549 CONTINUE
C
C      DATA CORRECTION
C
550 CONTINUE
WRITE BINARY(3) LT
CALL FCLOSE(3)
CALL MTDIO(1,10000K,IEM,IS,IERR)
666 STOP
701 STOP STRNF2 TAPE TABFIL
END

```

```

C*****
C
C      SUMCAT
C
C      THIS ROUTINE SUMS CLASSIFIED CATEGORIES INTO
C      GENERAL CATEGORIES FOR OUTPUT
C
C      SEQUENCE: SUMCAT INFIL
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS FAUST
C*****
C      DIMENSION COUNT(60), IEQ(60,20), NUM(20), ICAR(10,20)
C      DIMENSION JFIL(34), LSUM(20), PER(60), KFIL(34), ISWS(2)
C      CALL OPEN(1, "COM. CM", 1, IE)
C      CALL COMARG(1, JFIL, ISWS, IE)
C      CALL COMARG(1, JFIL, ISWS, IE)
C      CALL OPEN(3, JFIL, "B")
C      CALL COMARG(1, KFIL, ISWS, IE)
C      CALL FOPEN(2, KFIL, "B")
C      READ(3) ISUM
C      DO (L=1, ISUM)
101  :   READ (3,101) ( ICAR(J,L), J=1,10)
C      :   FORMAT (10A2)
C      :   READ(3) NUM(L)
C      :   NUM2=NUM(L)
C      :   READ(3) ( IEQ(J,L), J=1, NUM2)
C      :   ...FIN
C      READ(3) NWQ
C      DO (K=1, NWQ)
C      :   STOT=0.
C      :   DO (KG=1,60) COUNT(KG)=0.
130  :   READ(2,130) NA1
C      :   FORMAT(14)
131  :   WRITE(12,131) NA1
C      :   FORMAT(2X, "   WQMU #   ", I5///)
C      :   DO (KZ=1,60)
70   :   :   READ(2,70) COUNT(KZ), PER(KZ)
C      :   :   FORMAT(2X, F12.2, 2X, F12.2)
C      :   :   WRITE(10) COUNT(KZ), PER(KZ)
C      :   :   ...FIN
C      :   DO (N=1, ISUM)
C      :   :   NUM2=NUM(N)
C      :   :   S2=0
C      :   :   S1=0
C      :   :   DO (L=1, NUM2)
C      :   :   :   IQ= IEQ(L,N)
C      :   :   :   S1=S1+COUNT(IQ)
C      :   :   :   S2=S2+PER(IQ)
C      :   :   :   ...FIN
C      :   :   WRITE(12,102) N, ( ICAR(J,N), J=1,10), S1, S2
102  :   :   STOT=STOT+S1
C      :   :   FORMAT(2X, I2, 2X, 10A2, 2X, F10.1, "   ACRES", 2X, F6.3, "   PERCENT"
C      :   :   ...FIN
C      :   WRITE(12,103) STOT
103  :   :   FORMAT(6X, "TOTAL   ", 10X, F12.1, "   ACRES")
C      :   WRITE(12,200)
200  :   :   FORMAT(//////////)
C      :   ...FIN
C      STOP
C      END

```

```

C*****
C
C      SYMINV  (SUBROUTINE)
C
C      INVERTS A SYMMETRIC MATRIX
C
C*****
C
C      CREATED AT NASA JSC
C*****
C      SUBROUTINE SYMINV(A,AI,DET,N)
C      DIMENSION A(N,N),AI(N,N),QC(12),D(12),DI(12)
C      INTEGER R
C      DET=1.0
C      R=0
10    R=R+1
C      IF(R.GT.N)GO TO 19
C      QC(R)=1.0
C      I=0
20    I=I+1
C      IF(I.GT.R-1)GO TO 29
C      QC(I)=0.0
C      J=0
30    J=J+1
C      IF(J.GT.R-1)GO TO 20
C      QC(I)=QC(I)-AI(I,J)*A(J,R)
C      GO TO 30
29    CONTINUE
C      D(R)=A(R,R)
C      IF(D(R).LE.0.0)GO TO 52
C      K=0
40    K=K+1
C      IF(K.GT.R-1)GO TO 49
C      D(R)=D(R)+A(R,K)*QC(K)
C      GO TO 40
49    CONTINUE
C      DET=DET*D(R)
C      IF((D(R)/A(R,R)).LT.1.E-8)GO TO 52
51    DI(R)=1.0/D(R)
C      GO TO 60
52    DI(R)=0.0
1010  WRITE(12,1010)R,D(R),DET
60    FORMAT(9H AT STEP ,11,4H D =,E10.5,6H DET =,E10.5)
70    I=0
C      CONTINUE
C      I=I+1
C      IF(I.GT.R)GO TO 100
C      AI(R,I)=0.0
C      AI(I,R)=0.0
C      J=0
80    J=J+1
C      IF(J.GT.R)GO TO 70
C      QD=QC(I)*DI(R)
C      AI(I,J)=AI(I,J)+QD*QC(J)
C      GO TO 80
100   CONTINUE
C      GO TO 10
19    CONTINUE
C      RETURN
C      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      THRST  (SUBROUTINE)
C
C      PURPOSE      UPDATE THE MEAN AND VARIANCE OF THE THRESHOLD STATISTICS
C
C      DESCRIPTION OF PARAMETERS
C
C      INPUT
C      CALLING SEQUENCE
C      X1,V1,N1 - CURRENT MEAN, VARIANCE, AND WEIGHT
C      X2,N2 - VALUE AND WEIGHT OF NEW SAMPLES
C
C      OUTPUT
C      CALLING SEQUENCE
C      X1,V1 - UPDATED VALUES
C*****
C
C      CREATED AT NASA/JSC  (ASTEP)
C*****
C      SUBROUTINE THRST(X1,V1,N1,X2,N2)
C      REAL N1
C      IF(N1.GT.0.) GO TO 10
C      X1=X2
C      V1=0.
C      RETURN
C 10  X=(N1+FLOAT(N2))
C      R1=(N1)/X
C      R2=FLOAT(N2)/X
C      X=R1*X1+R2*X2
C      V1=R1*(V1+X1**2)+R2*(X2**2)-X**2
C      X1=X
C      RETURN
C      END
```



```

C*****
C
C      TOP02
C
C      READS CYBER CONVERTED TOPO TAPES AND
C      DISPLAYS THEM TO THE COMTAL.
C
C      SEQUENCE:  TOP02 MTU:F
C*****
C
C      CREATED AT GEORGIA TECH LES
C
C      PROGRAMMER:  MICHAEL D. FURMAN
C*****
C      INTEGER INP(1810), IWORK(512), IMAG(256), ITAPE(10), ISW(2)
C      CALL OPEN(1, "COM.CM", 1, IERR)
C      CALL COMARG(1, ITAPE, ISW, IERR)
C      CALL COMARG(1, ITAPE, ISW, IERR)
C      ACCEPT "FAST SCAN? (1=YES, 0=NO) ", IFSCAN
C      WHEN (IFSCAN.EQ.1) IFSCAN=4
C      ELSE IFSCAN=1
C      ACCEPT "SKIP RECORDS? ", ISKP
C      ACCEPT "MAX ACCEPTED PEAK VALUE (AFTER SUBTRACTION) ", IPEAK
C      P2= IPEAK/255.0
C      ACCEPT "ASSIGNED PEAK SHADE VALUE? ", ISHADE
C      ACCEPT "SUBTRACT WHAT VALUE FROM ALL DATA? ", ISUB
C      ACCEPT "BEGIN WITH ELEMENT? ", IEL
C      IEL=IEL+7
C      IEND= IEL+511
C      IF (IFSCAN.EQ.4) IEND=1800
C      IF (IEND.GT.1800) IEND=1800
C      CALL MTOPD(3, ITAPE, 0, IE)
C      LINE=ISKP
C      DO (I=1, ISKP) CALL MTDIO(3, 0, IMP, IS, IE, IC)
C      DO (M5=1, 256)
C      :   TYPE "LINE NUMBER ", LINE
C      :   LINE=LINE+IFSCAN
C      :   DO (I=1, IFSCAN) CALL MTDIO(3, 0, IMP, IS, IE, IC)
C      :   K=1
C      :   DO (I=IEL, IEND, IFSCAN)
C      :   :   IWORK(K)=IMP(I)
C      :   :   K=K+1
C      :   ...FIN
C      :   K2=K-1
C      :   SUBTRACT-CONSTANT-AND-CUT-OFF-PEAKS
C      :   CALL PAC8(IWORK, IMAG, 512)
C      :   CALL RIMWRITE(0, 0, IMAG, 256)
C      :   IF (M5.EQ.256) ACCEPT "TYPE A '1' TO CONTINUE ", IANS
C      :   IF (IANS.EQ.1) M5=1
C      ...FIN
C      STOP
C      TO SUBTRACT-CONSTANT-AND-CUT-OFF-PEAKS
C      :   DO (M6=1, K2)
C      :   :   WHEN (IWORK(M6).LT. ISUB) IWORK(M6)=0
C      :   :   ELSE IWORK(M6)=IWORK(M6)-ISUB
C      :   :   IF (IWORK(M6).GT. IPEAK) IWORK(M6)=IPEAK
C      :   :   IWORK(M6)=IWORK(M6)/P2
C      :   ...FIN
C      ...FIN
C      END

```

TRAIN3

THIS PROGRAM IS DESIGNED FOR TRAINING FIELD SELECTION
AND STATISTICS GENERATION FOR THREE IMAGES

SEQUENCE: TRAIN3 MTU:F

DESIGNED AT GEORGIA TECH EES

PROGRAMMERS: NICKOLAS L. FAUST
ROBERT A. MADDOX
MICHAEL D. FURMAN

DIMENSION V(4,4),XM(4),IV(101),IVX(101),IVY(101)
DIMENSION D(4,2),MX(4),IDEV(30)
INTEGER FIELD(17),ISWS(2)
COMMON /DTRANS/ IDAT(256),IXD(512,4),IDUM(512)
COMMON /HISTX/ ICOUNT(4,100)
COMMON /BUFFER/ IBUF(1700)

ORIGINAL PAGE IS
OF POOR QUALITY

OPEN-FILES

INIT-VARIABLES

FOREVER

```

: ASK-QUESTIONS
: CONDITIONAL
:   ( IFUN.EQ.1) READ-IMAGE
:   ( IFUN.EQ.2) ALARM-SCREEN
:   ( IFUN.EQ.3) MAGNIFY-SUBSET-OF-SCREEN
:   ( IFUN.EQ.4) TAKE-TRAINING-SAMPLE
:   ( IFUN.EQ.5) TAKE-GCP
:   ( IFUN.EQ.6) DIGITIZE-BOUNDARY
:   ( IFUN.EQ.7) CALL FSWAP("SCRIPTO.SV")
:   ( IFUN.EQ.8) CALL FSWAP("INK.SV")
:   ( IFUN.EQ.9) CALL FSWAP("WFMBGB.SV")
:   ( IFUN.EQ.10) CALL FSWAP("CHCOLR.SV")
:   ( IFUN.EQ.11) CALL FSWAP("FILTER2.SV")
:   ( IFUN.EQ.12) CALL FSWAP("CLYDE5.SV")
:   ( IFUN.EQ.98) BLANK-GRAPHICS
:   ( IFUN.EQ.99) CLOSE-FILES
:   ( OTHERWISE)
:   : TYPE "PLEASE ENTER ONE OF THE NUMBERS GIVEN"
:   : ...FIN
: ...FIN

```

TO OPEN-FILES

```

: CALL OPEN(1,"COM.CM",1,IFER)
: CALL COMARG(1,FIELD,ISWS,IFER)
: CALL COMARG(1,FIELD,ISWS,IFER)
: IF (FIELD(1).NE."IT")
:   : TYPE "PARAMETER ERROR TRAIN3 MTU:F"
:   : TYPE "TAPE NAME PROBABLY NOT ENTERED"
:   : STOP PARAMETER ERROR IN TRAIN3 MTU:F ERROR HALT
: ...FIN
: CALL MTOPD(2,FIELD,0,IFER)
: CALL FOPEN(3,"VERTEX","B")
: CALL FOPEN(4,"HISTS","B")
: CALL FOPEN(5,"SAVSIG","B")
: TYPE "INPUT TAPE NUMBER IN 612 FORMAT "
: READ(11,96) IT1,IT2,IT3,ID1,ID2,ID3
: FORMAT(612)
: WRITE BINARY(3) IT1,IT2,IT2,ID1,ID2,ID3
: WRITE BINARY(4) IT1,IT2,IT3,ID1,ID2,ID3
: WRITE BINARY(5) IT1,IT2,IT3,ID1,ID2,ID3
: CALL MTDIO(2,0,IBUF,IS,IFER)
: CALL MTDIO(2,0,IBUF,IS,IFER)
: ...FIN

```

CC
TO INIT-VARIABLES

: MODE=0
: MODE1=2
: MAG=1
: ICH=1
: ND=4
:...FIN

CC
TO ASK-QUESTIONS

: TYPE " "
: TYPE "THE FOLLOWING FUNCTIONS ARE AVAILABLE, SELECT BY NUMBER"
: TYPE " "
: TYPE " 1 - READ IMAGE 9 - USE WFM"
: TYPE " 2 - ALARM 10 - USE CHCOLR"
: TYPE " 3 - MAGNIFY 11 - USE FILTER2"
: TYPE " 4 - TRAINING 12 - USE CLYDE5"
: TYPE " 5 - GCP"
: TYPE " 6 - DIGITIZE"
: TYPE " 7 - USE SCRIPTO 98 - BLANK GRAPHICS"
: TYPE " 8 - USE INK 99 - STOP"
: TYPE " "
: ACCEPT "FUNCTION NUMBER ? ",IFUN
:...FIN

CC
TO TAKE-TRAINING-SAMPLE

: K=1
: ICLOSE=1
: PAUSE POSITION CURSOR FOR TRAINING FIELD THEN RETURN
: CALL GETXY(IXS,IYS)
: IXS=IAND(IXS,377K)
: IYS=IAND(IYS,377K)
: IVX(1)=IXS
: IVY(1)=IYS
: IXS1=IXS
: IYS1=IYS
: TYPE " CURSOR COORDS ",IXS,IYS
: TYPE " POSITION CURSOR AT NEXT VERTEX"
: ACCEPT " ENTER 1 TO READ POSITION, 2 TO CLOSE : ",ICLOSE
: REPEAT UNTIL (ICLOSE.NE.1)
: : CALL GETXY(IX,IY)
: : CHECK-XY
: : CALL VECTOR(1,IXS,IYS,IX,IY,1,1) ;DRAW ON GRAPHICS 1
: : TYPE " END PTS OF SEGMENT",IXS,IYS,IX,IY
: : K=K+1
: : IVX(K)=IX
: : IVY(K)=IY
: : IXS=IX
: : IYS=IY
: : ACCEPT " ENTER 1 TO READ POSITION , 2 TO CLOSE : ",ICLOSE
: : ...FIN
: CALL VECTOR(1,IXS,IYS,IXS1,IYS1,1,1)
: IYMAX=0
: IXMAX=0
: IXMIN=10000
: IYMIN=10000
: IVX(K+1)=IVX(1)
: IVY(K+1)=IVY(1)
: KP=K+1
: DO (IR=1,KP)
: : IVX(IR)=IVX(IR)/MAG+ICOL
: : IVY(IR)=IVY(IR)/MAG+IL
: : IF(IVY(IR).GT.IYMAX) IYMAX=IVY(IR)
: : IF(IVY(IR).LT.IYMIN) IYMIN=IVY(IR)
: : IF(IVX(IR).GT.IXMAX) IXMAX=IVX(IR)
: : IF(IVX(IR).LT.IXMIN) IXMIN=IVX(IR)
: : ...FIN

```

: ISTART=IYMIN
: IEND=IYMAX
: TYPE " BOX LIMITS", ISTART, IEND, IXMIN, IXMAX
: CALL RDATA( ISTART, IEND, K, IVX, IVY, IXD, NP)
: CALL HIST3B(NP,3) ;PUT HISTOGRAM ON GRAPHICS 3
: ACCEPT " INPUT 1 FOR STATS : ", ISTAT
25 : CONTINUE
: IF( ISTAT.EQ.1)
: : DO (J1=1,4)
: : : XM(J1)=0.0
: : : DO (J2=1,4)
: : : : V(J1,J2)=0.0
: : : : ...FIN
: : : ...FIN
: : : I=0
: : DO (LK=1,NP)
: : : DO (LJ=1,4)
: : : : MX(LJ)=IXD(LK,LJ)
: : : : ...FIN
: : : : TYPE LK
: : : : CALL SEQCOV(MX,4,V,XM,I,D)
: : : : ...FIN
: : : CALL MATPRT(XM,ND,ND,1,5HMEANS)
: : : CALL MATPRT(V,ND,ND,ND,3HCOV)
: : : TYPE " STATS CALCULATED FOR ",NP," POINTS"
: : : ACCEPT " DO U WISH TO SAVE THIS SIGNATURE?1=YES : ", ISIG
: : : IF( ISIG.EQ.1)
: : : : TYPE " INPUT 8 CHARACTER NAME FOR SIGNATURE"
: : : : READ(11,107) NA1,NA2,NA3,NA4
107 : : : : FORMAT(4A2)
: : : : WRITE BINARY(5) NA1,NA2,NA3,NA4
: : : : WRITE BINARY(5) NP,XM,V
: : : : WRITE BINARY(3) NA1,NA2,NA3,NA4
: : : : WRITE BINARY(3) K,IYMIN,IYMAX
: : : : DO (K1=1,K) WRITE BINARY(3) K1,IVX(K1),IVY(K1)
: : : : WRITE BINARY(4) NA1,NA2,NA3,NA4
: : : : WRITE BINARY(4) ICOUNT
: : : : ...FIN
: : : : ...FIN
: : : ACCEPT " HAS SIGNATURE BEEN TREATED AS U WISH?1=YES : ", ISAT
: : : IF( ISAT.NE.1)
: : : : ISTAT=1
: : : : GO TO 25
: : : : ...FIN
: : : ...FIN
C
C
: TO CLOSE-FILES
: : CALL FCLOSE(3)
: : CALL FCLOSE(5)
: : CALL FCLOSE(4)
: : STOP TRAIN3 MTU:F NORMAL EXIT
: : ...FIN
C
C
: TO CHECK-MAG-FACTOR
: : WHILE (MAG.NE.1.AND.MAG.NE.2.AND.MAG.NE.4.AND.MAG.NE.8)
: : : TYPE "INCORRECT MAG FACTOR"
: : : ACCEPT "ENTER MAG FACTOR (1,2,4, OR 8)",MAG
: : : ...FIN
: : ...FIN

```

```

C
C
TO READ-IMAGE
:   ILT=IL
:   ICOLT=ICOL
:   TYPE "INPUT TOP LEFT COORDS FOR IMAGE (COL,LINE) "
:   ACCEPT "   ENTER -1,-1 TO KEEP OLD COORDS ",ICOL,IL
:   IF (IL.EQ.-1)
:   :   IL=ILT
:   :   ICOL=ICOLT
:   ...FIN
:   ILS=IL
:   IES=ICOL
:   ACCEPT "MAGNIFICATION FACTOR (1,2,4, OR 3) ? ",MAG
:   CHECK-MAG-FACTOR
:   CALL RDIMG(IL,ICOL,ICH,MAG)
...FIN

```

```

C
C
TO ALARM-SCREEN
:   ACCEPT "ALARM TO WHICH OVERLAY (0-3) ",IOV
:   CALL ALRM2(IOV)
...FIN

```

```

C
C
TO MAGNIFY-SUBSET-OF-SCREEN
:   PAUSE POSITION CURSOR AT UPPER LEFT AND HIT RETURN
:   CALL GETXY(IX,IY)
:   CHECK-XY
:   ICOL=IX/MAG+IES
:   IL=IY/MAG+ILS
:   TYPE "SCREEN COORDS",IX,IY
:   TYPE "   TAPE COORDS",ICOL,IL
:   ACCEPT "MAG FACTOR (1,2,4, OR 3) ? ",MAG
:   CHECK-MAG-FACTOR
:   CALL RDIMG(IL,ICOL,ICH,MAG)
...FIN

```

```

C
C
TO TAKE-GCP
:   ACCEPT "INPUT 1 TO READ CURSOR POSITION ",IG
:   REPEAT UNTIL(IG.NE.1)
:   :   CALL GETXY(IX,IY)
:   :   CHECK-XY
:   :   TYPE "SCREEN COORDS",IX,IY
:   :   ICOLKT=IX/MAG+ICOL
:   :   ILKT=IY/MAG+IL
:   :   TYPE "   TAPE COORDS",ICOLKT,ILKT
:   :   ACCEPT "INPUT 1 TO READ CURSOR POSITION ",IG
:   ...FIN
...FIN

```

ORIGINAL PAGE IS
OF POOR QUALITY

C
C

```
TO DIGITIZE-BOUNDARY
: ACCEPT "DRAW BOUNDARY ON WHICH OVERLAY (0-3) ? ",IOV
: PAUSE POSITION CURSOR AT START AND HIT RETURN
: CALL GETXY(IX,IY)
: CHECK-XY
: TYPE "SCREEN COORDS",IX,IY
: ICOLS=IX/MAG+ICOL
: ILS2=IY/MAG+IL
: TYPE "TAPE COORDS",ICOLS,ILS2
: IX1=IX
: IY1=IY
: IXS=IX
: IYS=IY
: ACCEPT "ENTER: 1 TO READ POSITION, 2 TO CLOSE, 3 TO STOP ",ICLOSE
: REPEAT UNTIL(ICLOSE.NE.1)
: : CALL GETXY(IX,IY)
: : CHECK-XY
: : CALL VECTOR( IOV,IXS,IYS,IX,IY,1,1)
: : TYPE "LAST VERTEX AT"
: : TYPE "SCREEN COORDS",IX,IY
: : ICOLS=IX/MAG+ICOL
: : ILS2=IY/MAG+IL
: : TYPE "TAPE COORDS",ICOLS,ILS2
: : IXS=IX
: : IYS=IY
: : ACCEPT "ENTER: 1 TO READ POSITION, 2 TO CLOSE, 3 TO STOP ",ICL
: : ...FIN
: IF (ICLOSE.EQ.2) CALL VECTOR( IOV,IXS,IYS,IX1,IY1,1,1)
: ...FIN
```

C
C

```
TO CHECK-XY
: IX=IAND(IX,32767)
: IY=IAND(IY,32767)
: ...FIN
```

C
C

```
TO BLANK-GRAPHICS
: ACCEPT "WHICH OVERLAY TO BLANK ? (0-3) ",IGRN
: DO (I=1,16) IDUM(I)=0
: DO (I=0,255) CALL GWR(IGRN,I,IDUM,16)
: ...FIN
```

C
C

```
STOP
END
```

```

C*****
C      UNPAC1      (SUBROUTINE)
C      UNPACKS LANDSAT GODDARD FORMAT
C*****
C      "CREATED AT NASA/JSC      (ASTEP)
C*****
C      SUBROUTINE UNPAC1
C      INTEGER BUF
C      COMMON/BUFFER/BUF(3300)
C      COMMON/COL/JS,JF
C      CALL MTB10(4,0,BUF,ISTAT,IERR,NW)
C      IF(NW.GT.1750)WRITE(10,1)
C      WRITE(10,2)NW
2      FORMAT(2X,"# OF WORDS READ = ",I5)
1      FORMAT(2X,"ERROR IN UNPAC1")
      IM=1630
      JC=0
      DO 20 I=1,NW
      NM=I+IM
20     BUF(NM)=BUF(I)
      DO 3 J=JS,JF
      I=J-1
      L1=-16+(I+(I/2)*6)*8
      JC=JC+1
      DO 4 I=1,4
      IL=L1+I*16
      II=IL/16
      I2=IABS(IL-II*16)
      II=II+1
      IF(I2.EQ.8)GO TO 10
      II=ISHFT(BUF(II+IM),-8)
      GO TO 5
10     CONTINUE
      II=IAND(BUF(II+IM),377K)
5      CONTINUE
      KK=I+(JC-1)*4
      BUF(KK)=255-II
4      CONTINUE
3      CONTINUE
      RETURN
      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*****
C
C      UNPAC4  (SUBROUTINE)
C
C      UNPACKS PART OF LINE INSIDE POLYGON
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      SUBROUTINE UNPAC4( IBUF, JBUF, JL, JR, N, M)
C      DIMENSION JBUF(4, N), IBUF(M)
C      IF(JR.GT.810) JR=810
C      DO (K=1,4)
C      :   L=((JL-1)/2)*4+K
C      :   LAST=L+(JR-JL+1)*2
C      :   J1=1
C      :   DO (I=L, LAST, 4)
C      :   :   JBUF(K, J1)=ISHFT( IBUF(I), -8)
C      :   :   JBUF(K, J1+1)=LAND( IBUF(I), 377K)
C      :   :   J1=J1+2
C      :   ...FIN
C      ...FIN
C      RETURN
C      END
```



```

C*****
C
C      UPPLT  (SUBROUTINE)
C
C      PURPOSE
C          GENERATES PRIORITY LIST GIVEN WEIGHTS
C
C      DESCRIPTION OF PARAMETERS
C
C          INPUT
C              CALLING SEQUENCE
C                  NVG - WEIGHTS, I-TH VALUE IS NUMBER OF POINTS IN I-TH
C                      CLUSTER
C                  NVM - NUMBER OF ENTRIES IN NVG
C
C          OUTPUT
C              CALLING SEQUENCE
C                  PLIST - PRIORITY LIST, J-TH VALUE IS INDEX OF THE J-TH
C                          LARGEST ENTRY IN NVG
C*****
C      CREATED AT NASA/JSC  (ASTEP)
C*****
C      SUBROUTINE UPPLT(PLIST,NVG,NVM)
C      REAL NVG,N
C      INTEGER PLIST,TLIST
C      DIMENSION PLIST(NVM),NVG(NVM),TLIST(20)
C      DO 10 I=1,20
C10  TLIST(I)=0
C      DO 30 L=1,NVM
C      N=-1
C      DO 20 I=1,NVM
C      IF(TLIST(I).EQ.1) GO TO 20
C      IF(NVG(I).LE.N) GO TO 20
C      N=NVG(I)
C      J=I
C20  CONTINUE
C      TLIST(J)=1
C30  PLIST(L)=J
C      RETURN
C      END

```

```

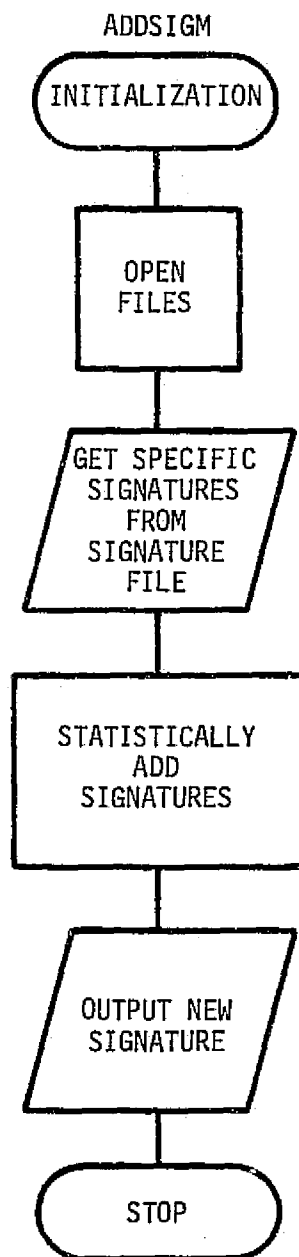
C*****
C
C      ZCOUNTY
C
C      CALCULATES PIXEL COORDINATES FOR ANY COUNTY PARTIALLY
C      OR WHOLLY CONTAINED IN A SCENE AND WRITES RESULTS TO DISK.
C
C      SEQUENCE:  NCOUNTY TAPE FILE
C*****
C
C      CREATED AT GEORGIA TECH EES
C
C      PROGRAMMER:  NICKOLAS L. FAUST
C*****
C      DIMENSION NCODE(50),ITEMP(2,300),FIELD(17),ISWS(2)
C      COMMON/DUM/IA(4100)
C      CALL OPEN(1,"COM.CM",1,IERR)
C      CALL COMARG(1,FIELD,ISWS,IER)
C      CALL COMARG(1,FIELD,ISWS,IER)
C      CALL MTOPD(2,FIELD,0,IE)
C      CALL COMARG(1,FIELD,ISWS,IER)
C      CALL FOPEN(3,FIELD,0,IE)
C      ISET=1
C      IOU=3
C      NCOD=0
C      ACCEPT "A1,A2,A3,B1,B2,B3 ",A1,A2,A3,B1,B2,B3
C      ACCEPT "SKIP RECORDS? ",ISK
C      IF (ISK.GT.0)
C      : DO (I=1,ISK) CALL MTDIO(2,0,IA,IS,IE,IC)
C      :...FIN
C      K=0
C      IV=0
C      IC=10
C      UNTIL (IC.LT.5)
C      : CALL MTDIO(2,0,IA,IS,IE,IC)
C      : TYPE " DATA ",(IA(KK),KK=1,IC)
C      : NCOUNTY=IA(1)
C      : YI=IA(2)*10000.+IA(3)
C      : XI=IA(4)*10000.+IA(5)
C      : IF(XI.LE..001.AND.YI.LE..001)STOP
C      : IV=IV+1
C      : IF(NCOUNTY.NE.NCOD)
C      : : ISET=1
C      : : JSET=0
C      : : KSET=0
C      : : IF(K.NE.0)
C      : : : NIV=IV-1
C      : : : DO (J=1,NIV)
C      : : : : IF(ITEMP(1,J).GT.0.AND.ITEMP(1,J).LT.4000)JSET=1
C      : : : : IF(ITEMP(2,J).GT.0.AND.ITEMP(2,J).LT.4000)KSET=1
C      : : : : IF(JSET.EQ.1.AND.KSET.EQ.1)ISET=0
C      : : : : JSET=0
C      : : : : KSET=0
C      : : : :...FIN
C      : : : IF(ISET.EQ.0)
C      : : : : NIV=IV-1
C      : : : : IV=1
C      : : : : WRITE(IOU)NCOD
C      : : : : WRITE(IOU)NIV
C      : : : : DO (K1=1,NIV)WRITE(IOU,200)ITEMP(1,K1),ITEMP(2,K1)
C      : : : : FORMAT(1X,2I5)
C      : : : : WRITE(IOU,300)NCOD
C      : : : : FORMAT(2X," END OF PIXELS FOR COUNTY = ",15)
C      : : : :...FIN
C      : : :...FIN
C      : : IV=1
C      : : NCOD=NCOUNTY
C      : : K=1
C      : : ISET=1
C      : :...FIN
C      : PY=A1+A2*XI+A3*YI
C      : PX=B1+B2*XI+B3*YI
C      : ITEMP(1,IV)=PX
C      : ITEMP(2,IV)=PY
C      :...FIN
C      STOP
C      END

```

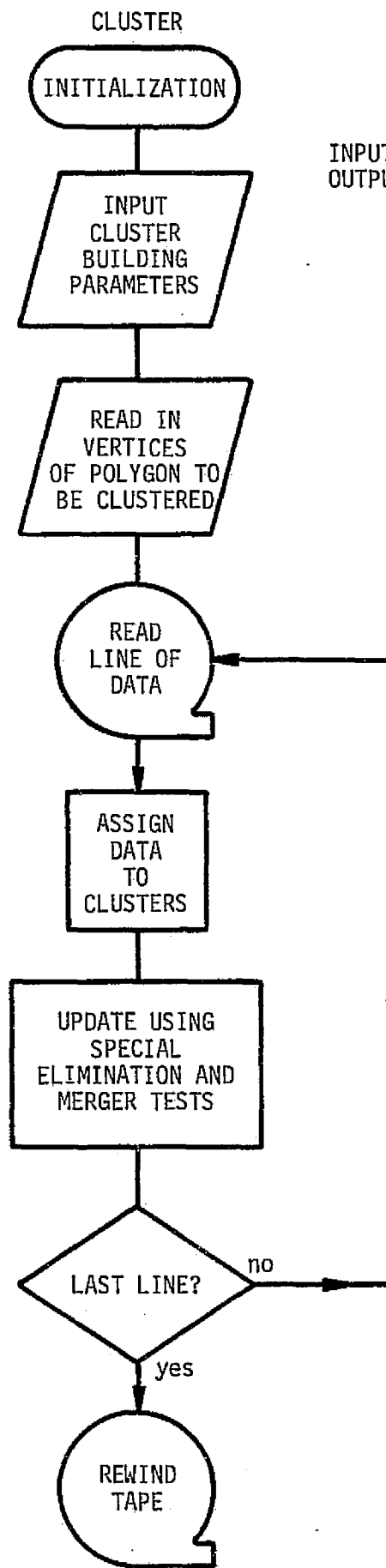
ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX C
MAIN PROGRAM FLOWCHARTS

ORIGINAL PAGE IS
OF POOR QUALITY

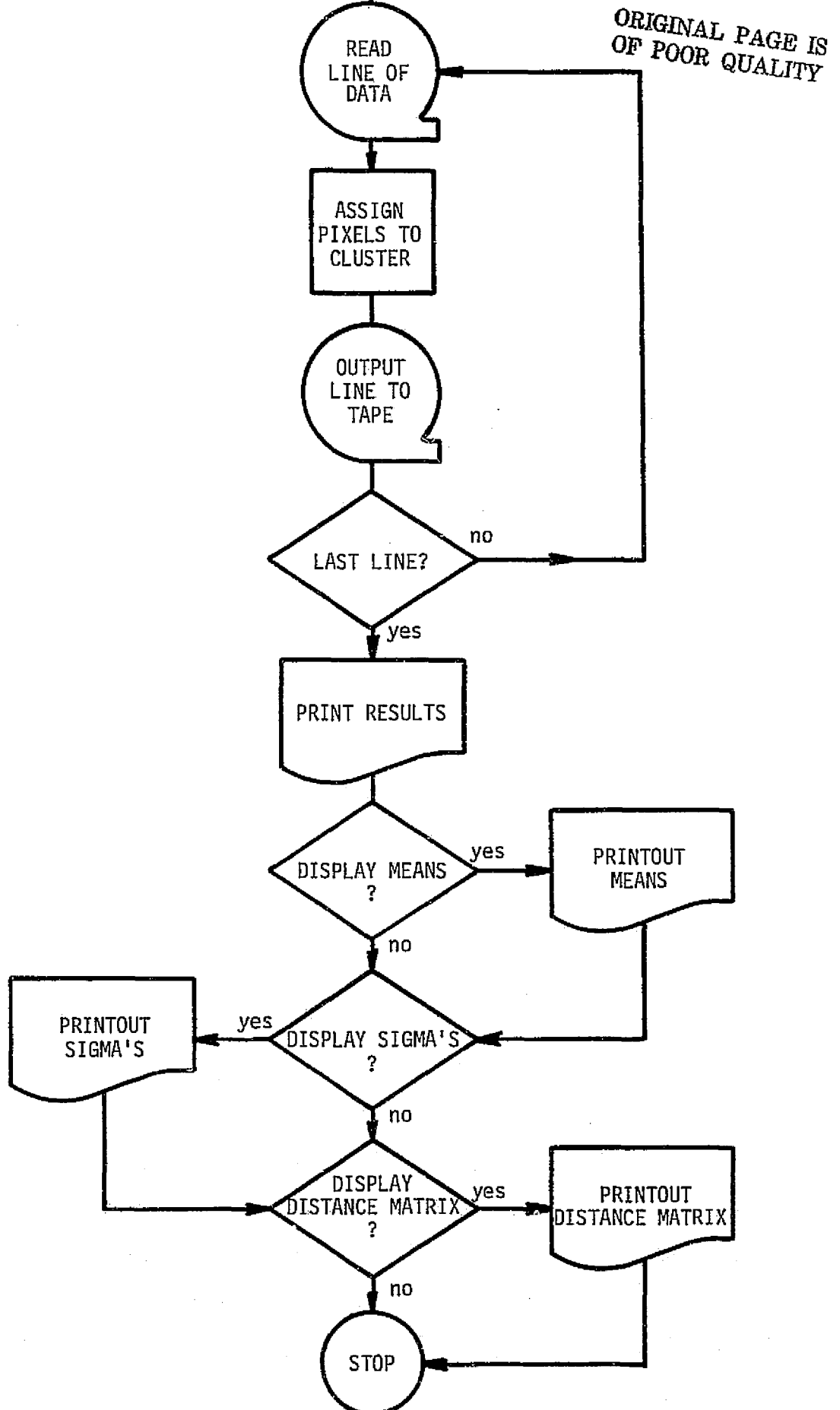


INPUT - 1) SIGNATURE NAME FILE
2) SIGNATURE FILE
OUTPUT - NEW BINARY
SIGNATURE FILE



INPUT - LANDSAT RAW DATA TAPE
OUTPUT - CLUSTERED DATA SET

CLUSTER (Continued)



LINCLASS

INITIALIZATION

INPUT - LANDSAT RAW DATA TAPE
OUTPUT - CLASSIFIED DATA TAPE

READ NUMBER
OF SIGNATURES
AND NAMES

CALL GSIG
GET
SIGNATURES

GET MEAN
AND CALCULATE
STANDARD DEVIATION

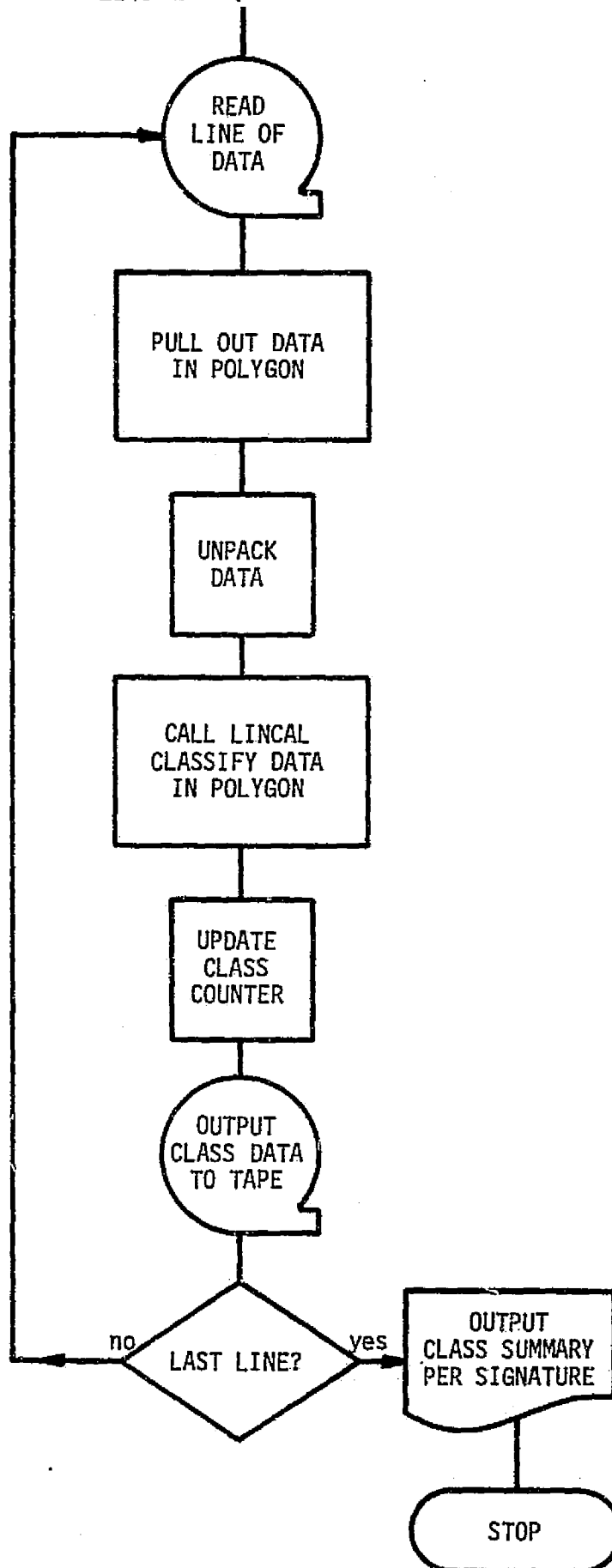
READ
POLYGON
BOUNDARIES

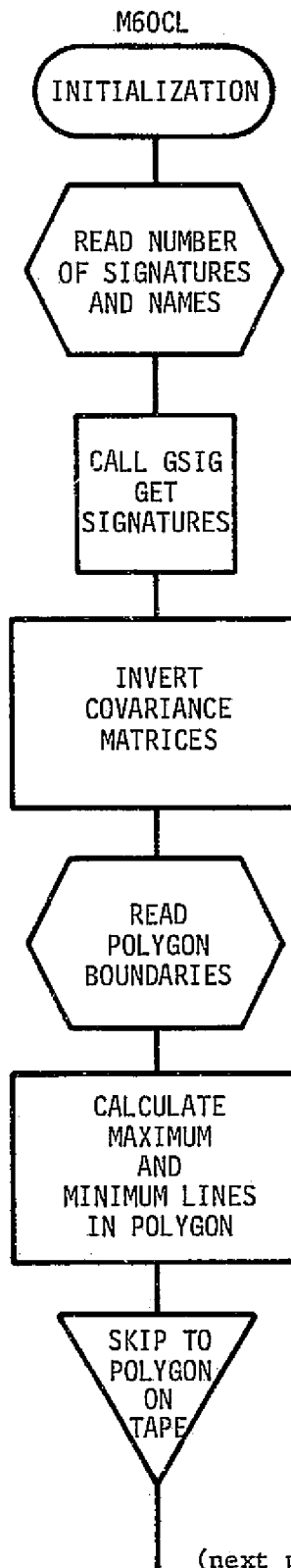
CALCULATE
MAXIMUM
AND
MINIMUM LINES
IN POLYGON

SKIP TO
POLYGON
ON
TAPE

(next page)

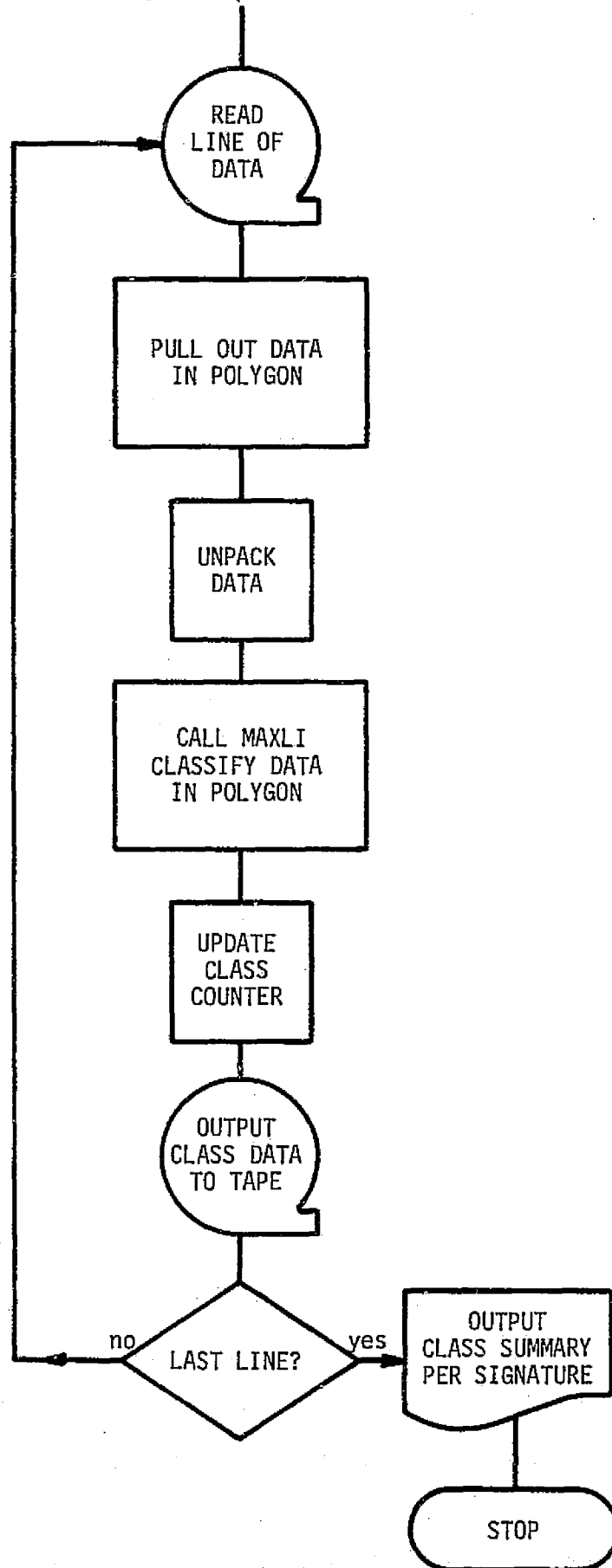
LINCLASS (Continued)

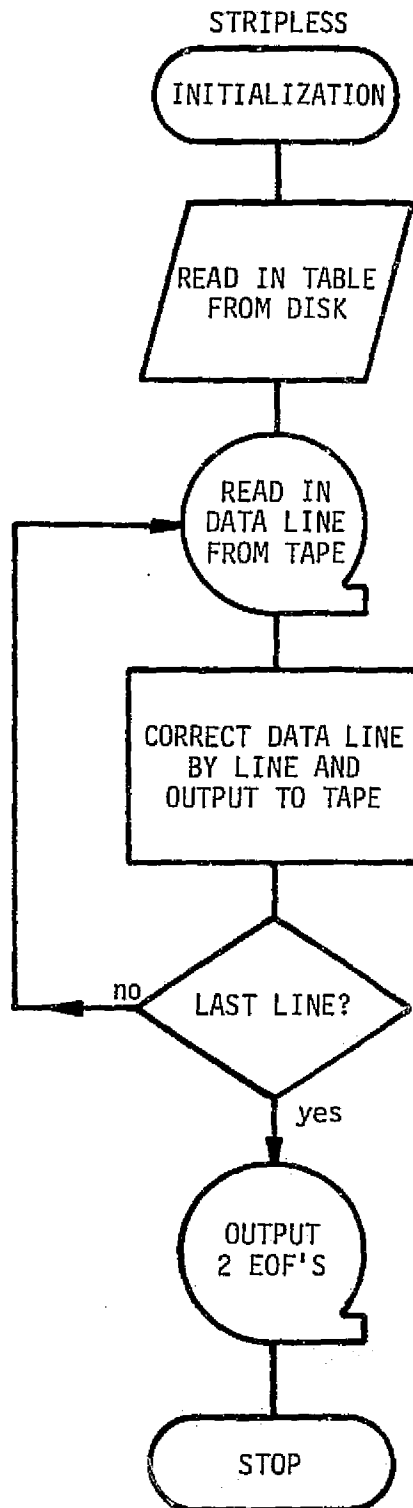




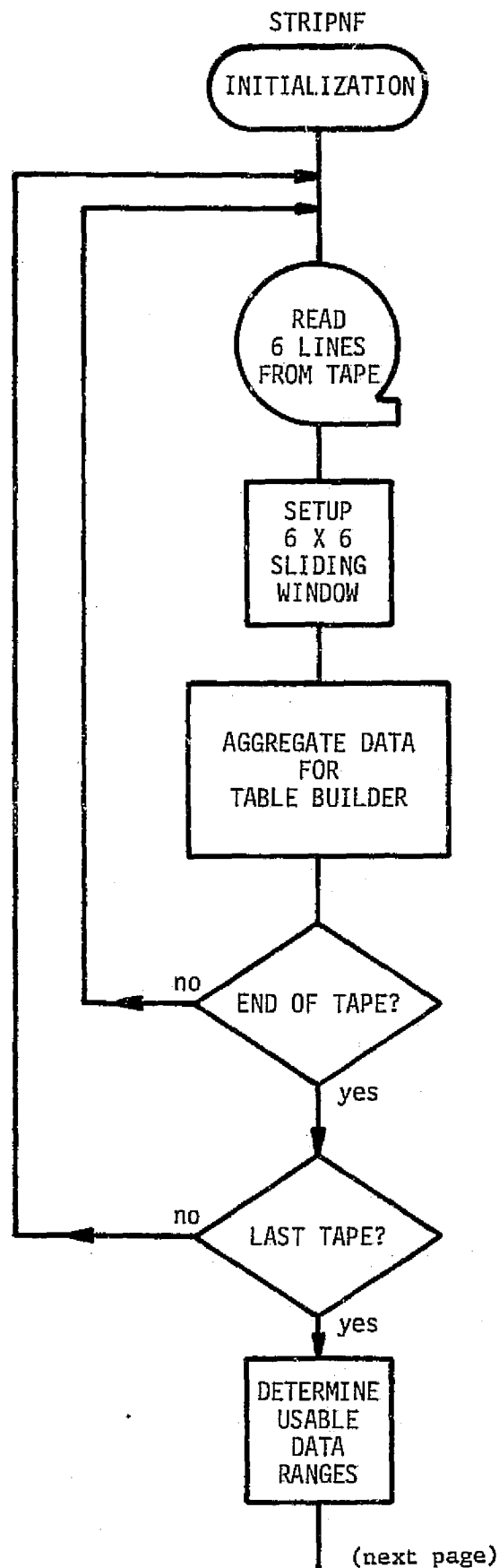
INPUT - LANDSAT RAW DATA TAPE
OUTPUT - CLASSIFIED DATA TAPE

M60CL (Continued)



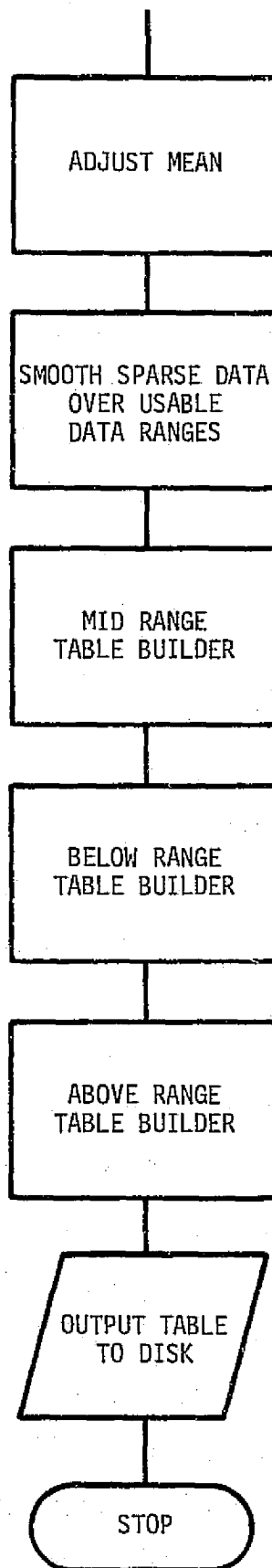


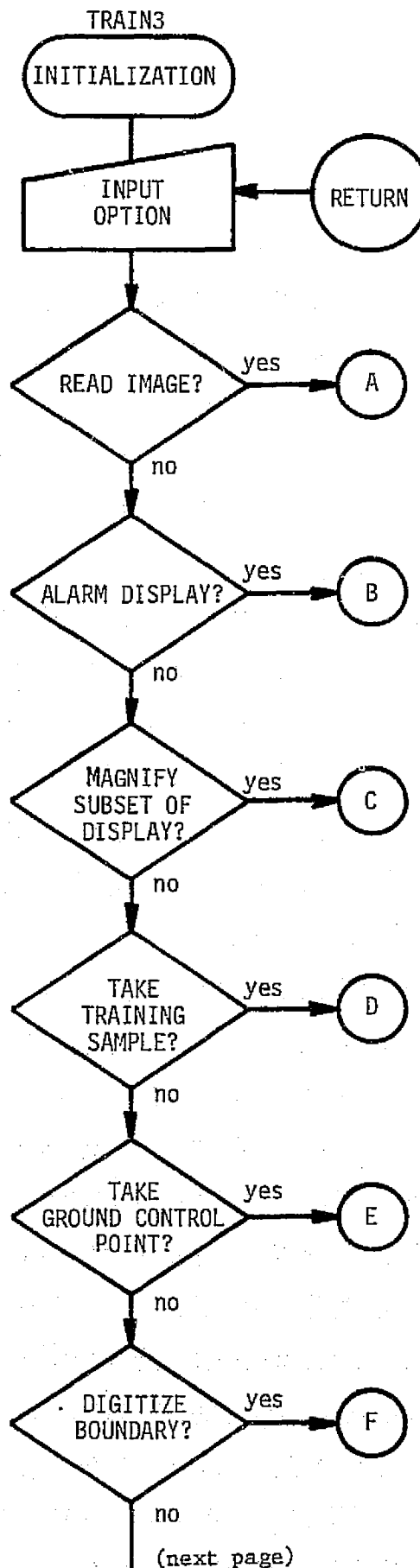
INPUT - DISK FILE
OUTPUT - DATA TAPE



INPUT - LANDSAT RAW DATA TAPE
OUTPUT - DISK FILE AND PRINTED

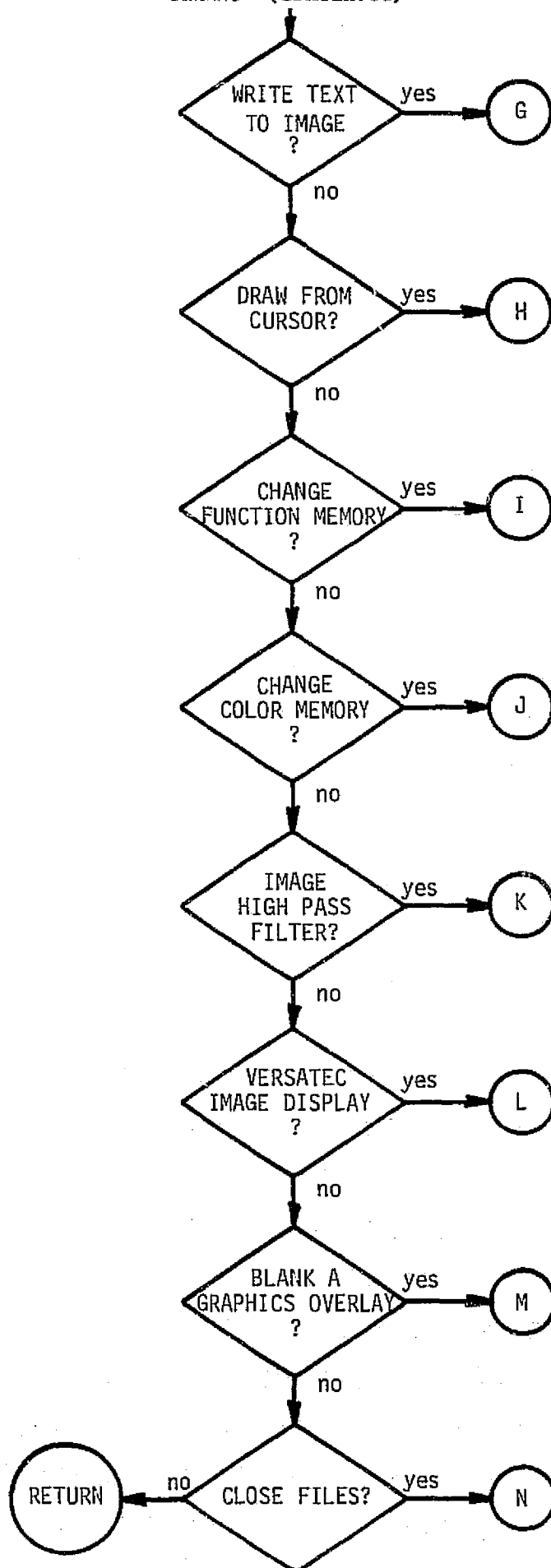
STRIPNF (Continued)



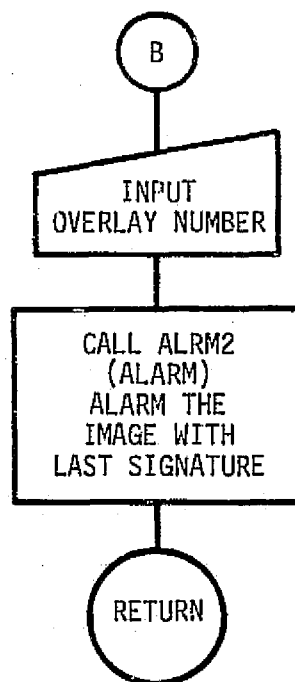
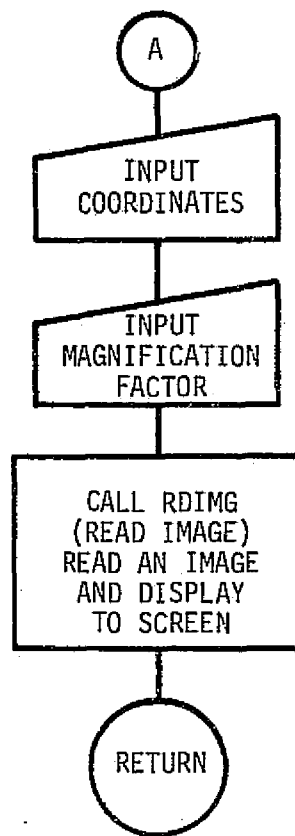


INPUT - LANDSAT RAW
DATA TAPE
OUTPUT - FILES FOR
a) SIGNATURES
b) VERTICES
c) HISTOGRAMS

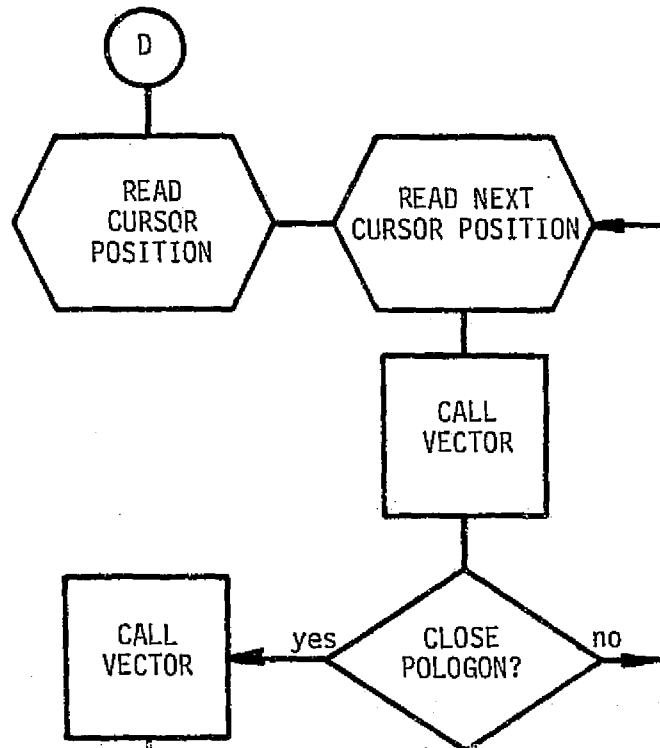
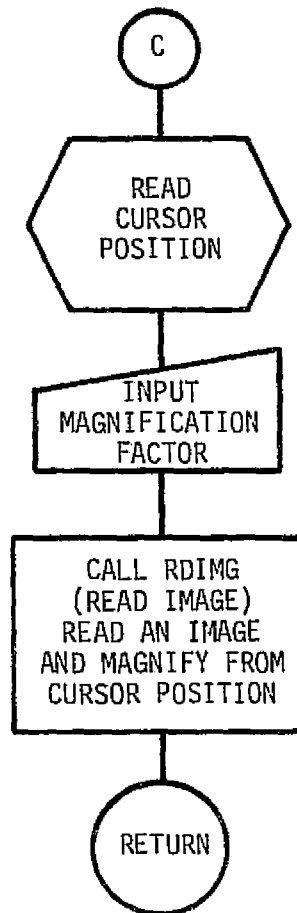
TRAIN3 (Continued)

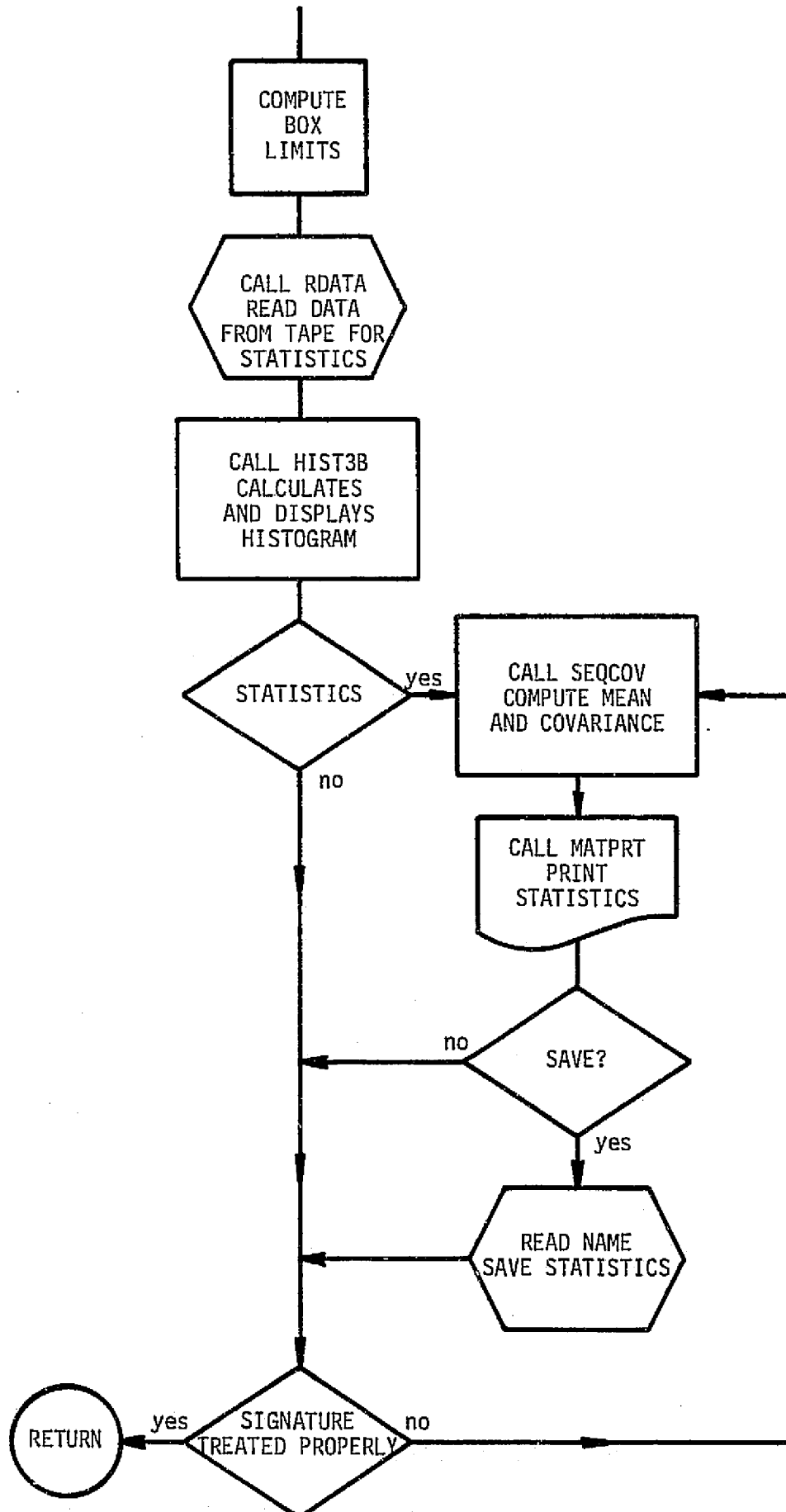


TRAIN3 (Continued)

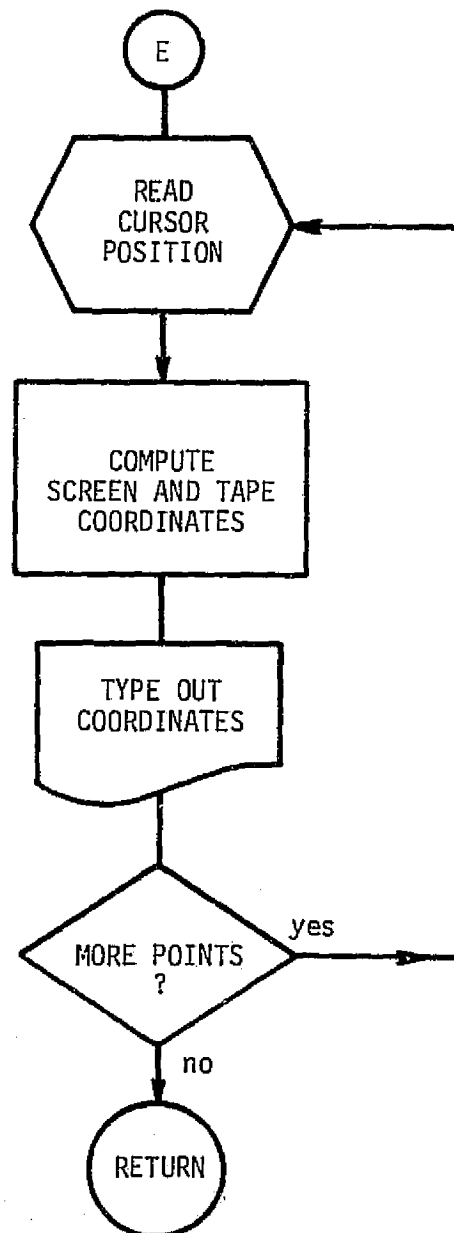


TRAIN3 (Continued)

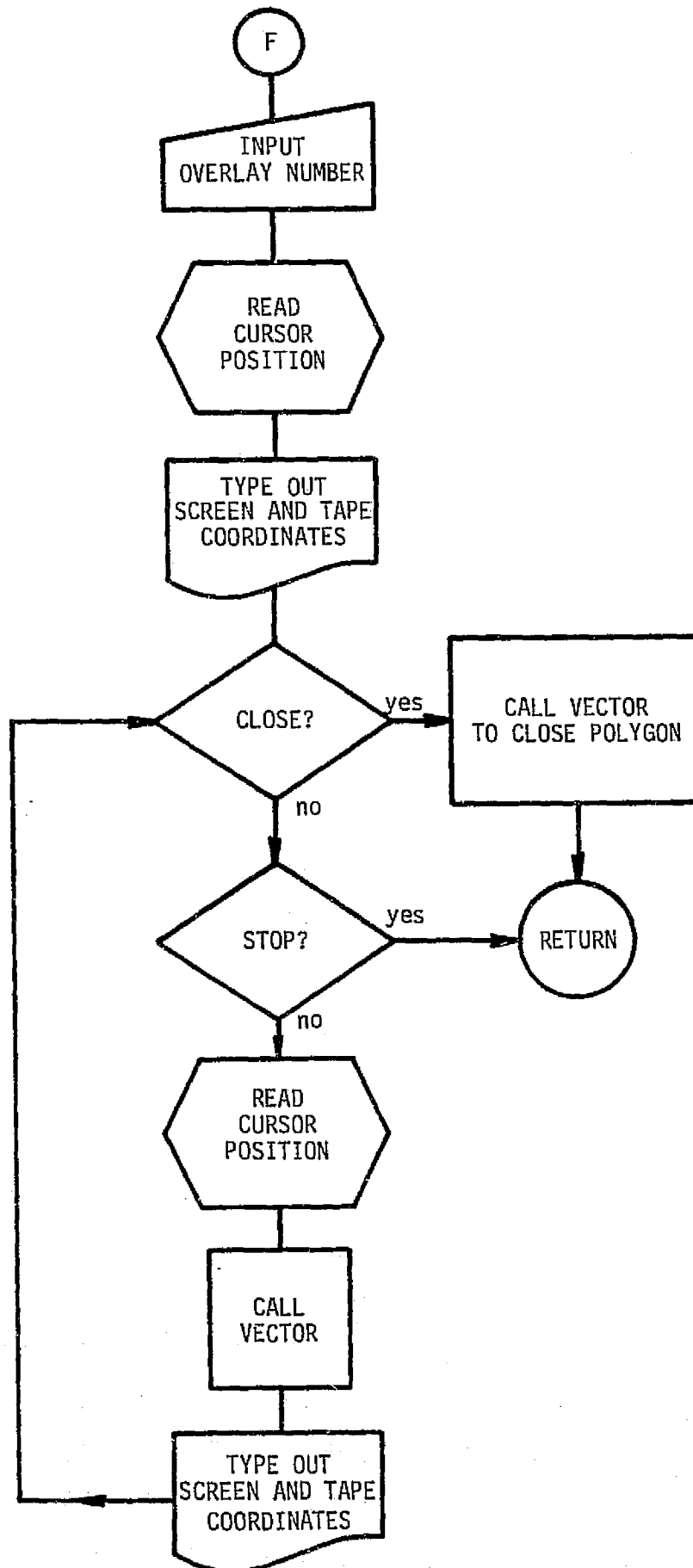




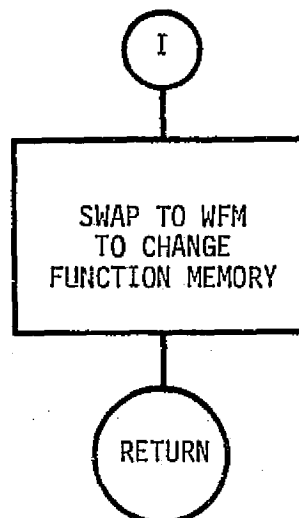
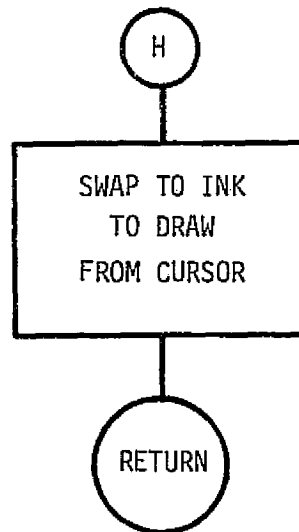
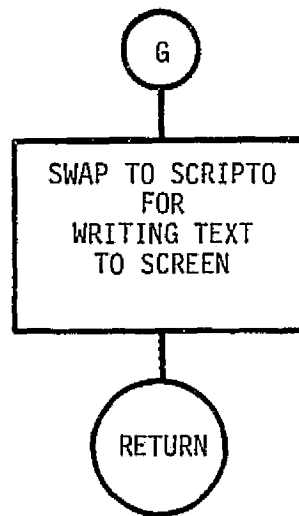
TRAIN3 (Continued)



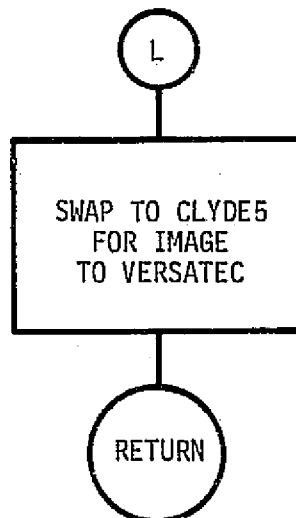
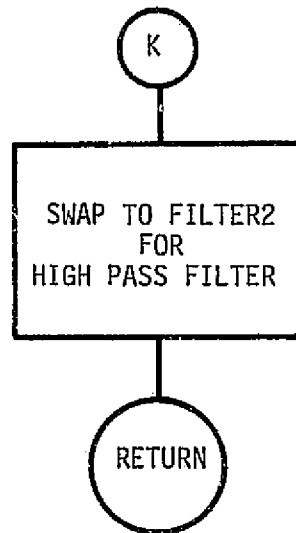
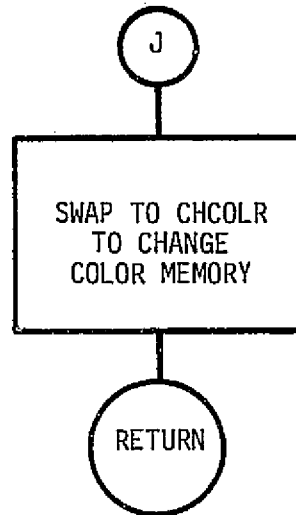
TRAIN3 (Continued)



TRAIN3 (Continued)



TRAIN3 (Continued)



TRAIN3 (Continued)

